Undergraduate Thesis
Submitted in the partial fulfilment
of the requirements for the award of

Bachelor of Technology in Computer Science and Engineering

MICROBIOLOGICAL DATA ANALYSIS WITH NON-LINEAR METHODS

Director: Dr. Luis Antonio Belanche Muñoz
Co-Director: Dr. Alfredo Vellido Alcacena

Author: Nikhith Sannidhi

Date: 25 June 2018

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

UPC

To my parents, who have always shaped and guided me on the right path and given me strength and motivational support to do different things. To my grandma, for her constant support. To my lovely sister, who keeps me smiling and happy all the time. To Dr. Luis Belanche and Dr. Alfredo Vellido without whom this accomplishment would have never been possible.

Abstract

Motivated to tackle the everlasting problem of water pollution, we consider the issue of water contamination in Europe provoked by fecal contamination. This research project uses a number of non-linear methods, mostly from the field of Machine Learning, for microbial source tracking, aiming to find the source of the fecal contamination. Data comprising 10,000 observations of water samples characterized by 45 variables, representing various microbial and chemical markers, is considered for the analysis as a classification problem. The non-linear Machine Learning techniques used are support vector machines and random forests. The mission is to find optimal discriminating parameters and use them to assess the best model for classification. The project kicks-off with a process that involves ranking the features using recursive feature elimination and then the ideal feature subset is selected with help of the Matthews correlation coefficient. We then use the obtained subset of features to train models, analyze their performances using Matthews correlation coefficient and F_1 score and select the preferred classification model for microbial source tracking. Results show that with a reduced number of features, both Support Vector Machine and Random Forest performed well with the Matthews correlation coefficient of over 0.9.

Resum

Motivats per abordar l'etern problema de la contaminació de l'aigua, abordem la problemàtica de l'aigua provocada per contaminació fecal a Europa. Aquest projecte de recerca utilitza una sèrie de mètodes no lineals, principalment del camp de l'Aprenentatge automàtic, per al rastreig de fonts microbianes, amb l'objectiu de trobar la font de contaminació fecal. Les dades comprenen 10.000 observacions de mostres d'aigua caracteritzades per 45 variables, que representen diversos marcadors microbians i químics, i es consideren per a un problema de classificació. Les tècniques d'aprenentatge automàtic no lineal utilitzades són màquines de vectors de suport i boscos aleatoris.

L'objectiu és trobar paràmetres de discriminació òptims i usar-los per avaluar el millor model per a la classificació. El projecte comença amb un procés que implica ordenar els marcadors mitjançant la seva eliminació recursiva i després es selecciona el subconjunt de marcadors ideals amb l'ajuda del coeficient de correlació de Matthews. Finalment es fan servir els subconjunts de característiques obtinguts per entrenar models, analitzar els seus rendiments i seleccionar el model de classificació preferit per al rastreig de fonts microbianes.

Resumen

Motivados por abordar el eterno problema de la contaminación del agua, abordamos la problemática del agua provocada por contaminación fecal en Europa. Este proyecto de investigación utiliza una serie de métodos no lineales, principalmente del campo del Aprendizaje automático, para el rastreo de fuentes microbianas, con el objetivo de encontrar la fuente de contaminación fecal. Los datos comprenden 10.000 observaciones de muestras de agua caracterizadas por 45 variables, que representan diversos marcadores microbianos y químicos, y se consideran para un problema de clasificación. Las técnicas de aprendizaje automático no lineal utilizadas son máquinas de vectores de soporte y bosques aleatorios.

El objetivo es encontrar parámetros de discriminación óptimos y usarlos para evaluar el mejor modelo para la clasificación. El proyecto comienza con un proceso que implica ordenar los marcadores mediante su eliminación recursiva y luego se selecciona el subconjunto de marcadores ideales con la ayuda del coeficiente de correlación de Matthews. Finalmente se usan los subconjuntos de características obtenidos para entrenar modelos, analizar sus desempeños y seleccionar el modelo de clasificación preferido para el rastreo de fuentes microbianas.

Acknowledgement

First of all, I would like to sincerely thank my home university, SASTRA Deemed University, Thanjavur, India, for offering me this wonderful opportunity of Semester Abroad Program. It is truly a once-in-a-lifetime opportunity for me and I am honoured to receive it. Apart from gaining knowledge on Machine Learning, I take back home loads of experiences, made memories with new friends and embraced the beauty of Barcelona. I would like to thank Barcelona School of Informatics, Universitat Politècnica de Catalunya for hosting me to pursue my final semester project at their prestigious school.

I would like to express my deepest gratitude to my project directors, Prof. Dr. Luis Belanche and Prof. Dr. Alfredo Vellido, for giving me valuable guidance and constructive feedback through the course of the project. They helped and motivated me to learn new concepts and programming languages. They provided me with excellent reference materials and papers. They supported and motivated me whenever I faced challenges in the project. I am profoundly grateful to both of them.

Special thanks to Prof. Marta Arias and Prof. Jaume Baixeries, for clarifying basic doubts in Machine Learning techniques and R language.

I would like to thank Prof. Dr. S. Vaidhyasubramaniam, Dean of Planning and Development, SASTRA Deemed University for organizing this program and Prof. Raja Subramanian, Office of International Relations, SASTRA Deemed University for helping me with the administrative and paperwork.

My heartfelt thanks to Gurudwara Nanaksar Sahib, Barcelona for extending their helping hand when in need of it. I thank my friends who made my stay fruitful and enjoyable.

Last, but never the least, I am thankful to my parents, Satyanarayana Sannidhi and Sunanda Sannidhi, my sister, Samhitha Sannidhi, and my grandmother, Prameela Alapati for their constant efforts and motivational support throughout my life.

Contents

List of Figures

List of Tables

Acronyms

MST Microbial Source Tracking

ML Machine Learning

ARA Antibiotic Resistance Analysis

SVM Support Vector Machine

RFE Recursive Feature Elimination

MCC Matthews Correlation Coefficient

CRAN Comprehensive R Archive Network

OVO One-Vs-One

OVA One-Vs-All

OOB Out-Of-Bag

RF Random Forests

Chapter 1

Introduction

In this chapter, we elaborate the context, describe the stakeholders and the goals of the project. We also discuss the design plan laid to carry out the project, as well as its economical aspects and sustainability.

1.1 Context and Problem Formulation

Technological advances have drastically changed the human lifestyle. Man has achieved countless wonders and broken many frontiers in terms of human success. But the other side of the coin is that this progress has of late threatened the natural environment, a trend that has accelerated over the last few decades. In the 21st century, we are seeing our environment degrade due to many of the human activities. The rise in average earth's temperature due to global warming is the epitome of today's ecological instability. The massive size of 8.9 million square miles of ozone hole (measured in 2016) alarms the world, wary of the negative effects of UV radiations exposure [1].

According to a study by the University of Maryland, 29.7 million hectares of vegetation on the planet has been destroyed in 2016 alone [2]. If we progress at the same rate, then we are creating a troublesome life for our children and for future generations. Therefore, it is very important to quickly realize the global issue and ensure that the human-activities drastically minimize their negative effects on nature.

From a generalized perspective, there are numerous causes for the environment deterioration. The contamination of water bodies is a huge problem in many cities and metropolitan urban areas and is one such cause for the depletion of the environment. This contaminated water is directly consumed by public for different purposes. The contaminated water mainly contains fecal coliforms. This is a major health concern as it may be the cause of serious health issues and can also be dangerous to life. There are various sources of origin of the fecal materials. These include human, domestic animals such as pets and livestock, wildlife and industrial waste. In order to mitigate and bring the contamination under control, it becomes very important

to ascertain the source of the fecal in a contaminated water body.

Spain houses more than 5000 km^2 of water bodies and has 4,964 km of coastline [3]. Water contamination of these water bodies is mainly due to human activities. After careful analysis of the root causes of water contamination, it has been concluded that the heavy pollution of water is due to fecal contamination coming from four sources, namely: human, pig, cow and poultry (chicken) [4]. The human population density of Spain is 93 persons/km^2 as of 2018 [5]. There are many farms that rear pigs, cows and poultry. The number of pigs in Spain, as of 2016, is 29.23 million heads, which stands the highest in Europe [6]. The number is so large that, as an illustration, the manure produced by these animals could fill up the famous FC Barcelona stadium 23 times over [7]. It is indeed an arduous and expensive task managing manure at this scale. Untreated waste is often directed to water bodies. Cattle and pigs together account to 70% of the total livestock units (LSU) in Spain [8] . Spain contributes 10.6% of the total poultry meat produced in the European Union - 28 [9].

Depending on the source, different mitigation measures can be taken. For example, if the water contamination is due to chicken fecal materials, then the poultry farms and slaughterhouses nearby the water body will be held responsible and necessary actions can be taken by the government to avoid further contamination, as the contamination mainly comes from these places. If the source is found out to be human fecal, a possibility source would be due to malfunctioning of a nearby waste water treatment plant (which could be due to heavy rains draining out the waste in the plant if the plant exceeds its capacity). Further, it should also be understood that the fecal contamination is harmful not only to the environment but pose a health risk to the public. Pig's waste can cause illness and human waste is dangerous and life-threatening. Thus, finding out the source of contamination is the topmost priority to solve the problem.

A method to figure out the source of contamination for given water sample is Microbial Source Tracking (MST). Microbial Source Tracking is the process of identifying the source of fecal contamination of water using microbial and chemical tracers. Figuring out the source of fecal contamination out of the four known sources can be understood, from a data analysis perspective, as a classification problem, amenable to supervised Machine Learning (ML) methods.

In this research activity, non-linear ML methods can be used to implement MST for a given water sample. Data used for the project is gathered from 5 countries of Europe, which are Spain, Finland, Portugal, Austria and the UK. This project experiments with the use of Support Vector Machines and Random Forests techniques and aims to find out the best model for correct classification. The project involves two experimental settings. One with 4 classes, namely: human, pig, cow and poultry and another with 2 classes, namely: human and non-human. The data set available to us for analysis contains many variables that are related microbial markers, which are specific to bacteriophages of either human, pig, cow, or poultry. The data set also contains

several derived variables which are built by taking the ratios among the above microbial markers. The novelty in the project is making use of this diverse data set (comprising of a variety of microbial indicators and 10,000 records) and performing separate analysis for the 4-class and 2-class settings. The reason why we perform the 2-class classification is that it is an important task from human context. Human fecal is very dangerous and can be fatal. If the water sample is classified as human, then it is an alarming situation and the community should act immediately to stop further contamination. Non-human fecal contamination is less dangerous. We perform the 4-class classification to estimate the specific source of fecal contamination. This classification precisely tells which animal, in the 'non-human 'category, has contaminated the water sample. In every experiment, we first select the best feature subset using Recursive Feature Elimination (RFE) and Matthews Correlation Coefficient (MCC). We then, use this feature subset to train the aforementioned non-linear classifiers and assess them using MCC and F1 score.

Experiments show that in the given data set comprising of 45 variables, many variables were found to be redundant. Both Support Vector Machine and Random Forest techniques showcases and MCC of over 0.9 with less than or equal to 17 variables. The selected feature subsets are presented in chapter 4 and their performance results are presented in chapter 5. The feature subsets are selected based on the performance of each feature subset with its size. Now that the essential feature subsets are known, instead of obtaining all the indicators, the microbiologists can now focus on investing time and money in obtaining the proposed feature subsets for MST. This will significantly save both time and money for the microbiologists.

1.2 Stakeholders

In this section, we discuss the relevant stakeholders i.e., the various people involved and benefited from this project.

1.2.1 Users

The developed product will be used by the microbiological researchers. They take water samples from contaminated water bodies, extract the chemical and microbe traces and concentration values from it. These values are then classified into one of the four possible sources of fecal contamination and this classification is achieved by the developed product, which is the outcome of this project.

1.2.2 Beneficiary

Once the source of fecal contamination in a water body is known, the results are reported to the government authorities who can take necessary steps to immediately stop and prevent further contamination of the water body. Therefore, the results will help the concerned government authorities to identify the people responsible for the contamination and sanction them.

1.2.3 Possible groups of people impacted by this project

As mentioned in the context, there are 4 sources of fecal contamination and the people responsible for each is given in table 1.1.

Source of Fecal Contamination	People Responsible for the fecal contamination(who reside/work nearby the water body)
Human fecal	Residence area or human settlement
Pig fecal	People working at pig farms
Cow fecal	People working at cow farms
Poultry fecal	People working at poultry farms

Table 1.1: Possible impacted people based on fecal source

1.2.4 Project Developers

This project is carried out under the valuable guidance of Dr. Luis Antonio Belanche Muñoz and Dr. Alfredo Vellido Alcacena, of the Soft Computing Research Group (SOCO) at the Department of Computer Science, Universitat Politècnica de Catalunya (UPC BarcelonaTech) in Barcelona, Spain.

1.3 State of the Art

A literature survey reveals state-of-art techniques to figure out the source of fecal contamination. To monitor the performance of the septic systems installed on-site, the Fluorescent Dye Tracing technique is often used. Positive dye results indicate the presence of fecal coliform bacteria in the water [10]. But this method demands heavy sampling procedure. Hagedorn *et al.* analyzed the patterns of antibiotic resistance in fecal streptococci to figure out the source of fecal in the contaminated water [11]. The results showcased a significant reduction of fecal coliform population in contaminated water when the cattle access to the watershed was restricted.

Ohad *et al.* used qualitative Polymerase Chain Reaction (qPCR) to discriminate fecal sources and to evaluate Karst spring susceptibilities. Three springs, which are geographically located nearby, were considered in the research. The data from these springs were collected both on a daily and weekly basis. Their research revealed that all three springs had non-identical MST profiles although they are in proximity to each other. They also stressed the importance of having carefully designed samples [12].

Wang *et al.* proposed a statistical model based on the principle of total probability for finding out the origin of fecal contamination in water samples taken in the United States. A Monte Carlo method was applied to the samples and the model was validated. The model output yields satisfying results and showed above 92% true classification [13].

Graves *et al.* used Antibiotic Resistance Analysis (ARA) method to implement the MST. The experiment was performed to assess how the water quality got affected by cattle. ARA results showcase that 60% of the contaminated water samples were dominated by cattle whereas deer, geese and human were minor contributors of fecal contamination [14].

Belanche *et al.* used ML methods to implement the MST. The goals were to figure out a subset containing least number of variables that have a high capacity of classification and use this subset of variables to build the models for classification. The methods used were Discriminant Analysis, Nearest Neighbor and Artificial Neural Networks [4].

1.4 Objectives

Within the global issue on environment management, the outcome of the project will be that of helping to reduce water pollution. Achieving this itself will involve activities from multiple disciplines and involved institutions but, in this project, we primarily focus on the computer science facets to solving the problem. The project is designed to achieve the following specific goals:

1. First, to explore the received microbiological data and look for missing values and outliers. Also to understand how the data, in each feature, is spread by constructing box plots.
2. To get a bird's-eye view of a few feature selection methods available and choose the appropriate one that is suitable and is within the project limits. We also make use of the MCC measure to understand how performance varies with the size of the feature subset by building graphs.
3. To understand and use the non-linear ML methods for classification by training the classifiers and testing them on test data.

4. Finally, to make use of the model measures to assess the classifier's performance and adjudge the preferred model that can be adopted for classification of microbial source of fecal contamination.

We aim to achieve the above objectives by performing four independent data analyses, described as follows:

1. Multiclass classification using single and derived features.
2. Multiclass classification using single markers only.
3. Binary classification using single and derived features.
4. Binary classification using single markers only.

The single features are a mixture of microbial, chemical and molecular parameters and the derived features are the ratios taken among the single features.

1.5 Project Planning

Just as the saying goes, "*a goal without a plan is just a wish*", it is vital and important to spend time and work on a carefully designed work plan. The duration of the project is only four months starting February 2018 and ending in June 2018. The project is carried out in the following phases, along with its description:

- **Knowledge Acquisition:** Before working in the modeling itself, it is important to acquire the necessary knowledge on the working mechanisms of the ML models. The parameters involved and the working of the considered non-linear methods will studied and understood [15]. Knowledge of R programming tools and the packages used for applying the ML models will be acquired as part of the project development [16].
- **Data Quality Assessment:** In this phase, we explore the given data. Using an R script, we identify all records which had zero values in all it features. The records that have records with all zeros, but have a class label make no sense for classification and so they will be removed. We will also construct and study the box plots of each feature.
- **Feature Selection:** In this phase, we survey a few feature selection algorithms and make a minor change, which is discussed in detail in the 3rd chapter. We use RFE to produce a rank list of features and find the appropriate size of feature subset using the MCC measure.
- **Refinement and Validation of Classifiers:** We apply Support Vector Machines and Random Forest models by making use of the R packages available at the Comprehensive R Archive Network (CRAN) [17]. We use 2-fold cross-validation to find the best parameters for the model.

- **Comparison of Classifiers:** In this phase, we tabulate the performance results of the two classifiers and discuss the selection of the best classifier. The two classifiers will be assessed using MCC and F1 score.

- **Reporting:** We document our activities and findings in the project report in this phase. The observations will be tabulated and the inferences and conclusions will be discussed. The possibility of future work stemming from this project will also be discussed at the end of the report.

1.6 Economic Budget

Budget plays an important role in the development of a project. In this computer science project, the entire expenditure incurred can be categorized into three sections, namely: hardware, software and human resources. The costs incurred in each of these sections are tabulated below:

- **Human Resources**
  Including the costs incurred for the labour work involved in the project. The developer will be considered as salaried on an hourly basis and a fixed cost per hour is assigned to each phase of the project. From the estimated time dedicated to the project that was calculated previously, we compute the expenditure incurred on human resources. Please note that these costs are fictional as the project is not developed for business purposes.

S.No.	Project Phase	Hours spent (in hours)	Cost per hour (in €)	Total cost of the phase (in €)
1	Knowledge Acquisition	100	15	1500
2	Quality Assessment of the Data	60	15	900
3	Feature Selection	90	15	1350
4	Refinement and Validation of Classifiers	100	20	2000
5	Comparison of Classifiers	40	15	600
6	Reporting	80	15	1200
	Total			7550

Table 1.2: Estimated Costs in Human Resources

- **Hardware**

  The hardware required for a data analysis project includes a high-end computer (or a laptop) with high-end RAM specifications for faster storage and processing. An optimal code structure would play an important role in reducing the hardware costs. In our project, the costly 10-fold cross-validation method takes a very long time to execute on an ordinary laptop and, consequently, demands a high-end system or compute cluster, thereby incurring excess costs for equipment. But, since the data we used is big enough, the use the 2-fold cross validation method, which executes in acceptable time, is sufficient. Therefore, the selection of the appropriate validation technique played an important role in reducing the hardware costs.

S.No.	Component	Useful life (in years)	Total cost (in €)	Amortized cost(in €)
1	Laptop	4	500	52,08
2	8 GB RAM Extension	4	100	10,41
	Total			62,49

Table 1.3: Estimated Costs in Hardware Section

- **Software**

  The software for the data analysis project involved the use of sophisticated data analysis tools such as Rstudio. Rstudio and the R environment do not incur any cost to the project budget, as they are open source. We also document the results in latex by using the free version of online latex editor, Overleaf.

S.No.	Component	Useful life (in years)	Total cost (in €)	Amortized cost(in €)
1	Rstudio	5	0	0
2	R Packages	5	0	0
3	Overleaf (online latex editor)	1	0	0
	Total			0

Table 1.4: Estimated Costs in Software

- **Total Costs**

  The estimated budget expenditure for the entire and successful completion of the project is 7,612.49€. This cost includes hardware, software and human resources and are summarized in Table 1.5.

S.No.	Component	Total cost (in €)
1	Hardware	62,49
2	Software	0
3	Human Resources	7.550
	Total	7.612,49

Table 1.5: Total Economic Budget

1.7 Sustainability

When outlining a project, one must not only aim at fostering the present conditions but also ensure that the future generations are not being affected. The future should not be compromised for the needs of the present. Therefore, it is essential to have a study of the sustainability of the project. The sustainability of this project is studied in three dimensions, namely: environmental, economic and social sustainability.

1.7.1 Environmental Sustainability

The entire project is executed on a laptop and has not involved any other heavy equipment or materials. The laptop consumes electrical power and releases carbon dioxide to the environment. According to technical specifications, the laptop uses 100 Watt per hour [18]. The dedicated time for executing the project is 470 hours. Therefore, the total energy consumed is 47 kWh. The electricity for Barcelona is received from the Barcelona Power Station, which uses Natural Gas as fuel [19]. One kWh produced by this natural gas burning plant generates nearly 0.94 kg of carbon dioxide emissions [20]. So, the carbon footprint of the project is approximately 44.18 kg. This is the estimated total amount of carbon dioxide released into the atmosphere during the entire 5-month course of the project and the amount is below the permissible limit. Efforts have been taken to develop an optimized code involving efficient Data Structures and code design to reduce the power consumed for executing the program. We can also assure that there will be almost NO environmental risks posed by this project and is environmentally sustainable. In fact, the whole idea of executing this project is improving the environmental sustainability, as it involves analyzing the fecal contamination in water bodies and identifying the potential actors who are responsible for the contamination. This can help the government take strict measures and actions to minimize and prevent water contamination and thus making a feasible and stable environment.

1.7.2 Economic Sustainability

The expenditure that is incurred for the project is 7,612.49€, as stated under section 1.5 of this document. Fortunately, this project does not involve any complex hardware, heavy materials, hazardous or expensive materials. The entire project is implemented on the computer and the software and packages used in the Rstudio software are open source and hence free of cost, thereby reducing the overall estimated costs. Therefore, the project is economically sustainable. To the best of my belief, the costs mentioned are the nominal costs of the products with no compromise in quality. As an optimist, efforts have been taken to reduce costs wherever possible. If the data set obtained is not sufficient to train a model to get the desired the accuracy, then we may need to obtain more data set from the client. This was the only potential economic risk. To avoid this, ML models that perform well with less training data (such as Support Vector Machines) were given preference in the selecting the model for classification. Fortunately, the data received is abundant and enough to train the models. When we searched for software for the project, we first looked for open-source software before looking for the licensed ones. Fortunately, we were able to find open-source software tools for executing the different parts of the project.

1.7.3 Social Sustainability

Today, we humans are playing a pivotal role in setting up an environment that may not be livable in future. Environmental deterioration poses health risks to human life and biodiversity. In order to make the world a better place to live, it is time for us to change and to prevent further deterioration of the environment consequently improving public health conditions. This project is aimed to contribute to preventing environmental degradation. Given a sample of contaminated water, this project predicts the source of the fecal contamination. The results will be given to the responsible government authorities and microbiologists. After ascertaining the results with the microbiologists, the authorities can then take further legal action against the responsible actors. If the sample is classified as human fecal, then the waster water treatment plant or the residential area nearby the contaminated water body, from which the water sample is taken, may be responsible. In all the other cases, the people rearing the animal (cow, pig and poultry) and the affiliated farms will be held responsible and necessary steps and legal actions will be taken against them by the government in order to prevent further contamination of water. The government authorities, the general public, the environmental activists and ultimately the environment will be beneficial by this project. But the actors held responsible for the contamination can be impacted heavily if the contamination is harsh and beyond the limits. Since this is a public health issue, these responsible actors may have to change their business strategies or methodology to prevent further pollution of water. But, if we want to change the environment, we need to change ourselves. Don't you agree?

Chapter 2

Background Knowledge

In this chapter, we provide information on the methods and techniques used to address the environmental problem presented in the previous chapter. This includes details on the feature selection techniques used, the supervised models adopted for classification and the measures used to assess the results obtained with these classifiers.

2.1 Dimensionality Reduction through Feature Selection

In any data analysis project, the given data consist of a number of predictors variables to be modelled. But the backdrop is that too many predictors (or variables or features) will slow down the training process. Also, some features can be redundant. For example, lets assume a data set has 'Date of Birth' and 'Age' as predictor variables. We know that age of a person can be calculated if the date of birth is known. Therefore, we see that 'age' is a redundant variable in the presence of 'date of birth' variable. Hence, we can safely remove age from our further analysis and this would not only speed up the training process but also eliminates the need of maintaining the 'age' variable for the client. Furthermore, some features can be completely unrelated to the model outcome information. In the worst case scenario, not only these features will not help in the modeling process, but might even hamper it.

A large number of features/variables (or a high data dimensionality from a spatial viewpoint) may invite an increased number of modeling problems. When combined with a large number of observations, computational times may become prohibitive; when combined with a small number of observations, the danger of overfitting the data sample is enhanced. Additional costs would be incurred if the current storage specifications do not meet the problem requirements. Moreover, it becomes very difficult to plot and visualize data of such high dimensionality. The problems and challenges faced due to the high dimensionality are commonly known as the *curse of dimensionality* [21].

In ML, dimensionality reduction aims at reducing the number of features used as predictors as given to the learning algorithm. The resultant predictor set, obtained from dimensionality

reduction, is used for training the model, or the selection is embedded in the training process itself. There are different groups of techniques to reduce the number of variables to train the classifiers and escape the curse of dimensionality. They include Feature Extraction and Feature Selection, described below

2.1.1 Feature Extraction

Feature Extraction techniques deal with combining the existing variables in such a way that the resulting set of extracted new variables does not compromise the model's performance. Thus, this reduces the overall number of variables into fewer combined variables. For Example, in Principal Component Analysis, if n is the number of variables, the algorithm looks for k orthogonal vectors such that $k \leq n$. The resultant orthogonal vectors are known as "Principal Components" and they are linear combinations of the original variables that account for as much variance in the data set as possible. These principal components are meant to essentially inherit the sense and insights of all the variables. Thereby, with Principal Component Analysis, we get new and a lesser number of features, while trying to avoid a negative impact on model performance [22].

2.1.2 Feature Selection

Feature Selection techniques in classification involve selecting a subset of those features that are the most "relevant" for such classification problem. That global subset relevance can be gauged using a statistical measure, or metrics such as accuracy or MCC. There are three different working principles of feature selection algorithms. These are:

- Filter Approach
- Wrapper Approach
- Embedded Approach

Filter Approach

In this family of methods, a selection algorithm takes in the complete set of features and outputs the best feature subset. The selection algorithm uses a criterion to figure out the best features. This criterion, for instance, can be information gain, variance, or correlation between the variables and the target variable. The higher the correlation between a feature X and the target variable Y, the better predictor for classification is X. These criterion values can be computed from the given data set. The resultant feature subset is then passed on to the learning algorithm. The drawback of the filter approach is that it selects the subset irrespective of the model considered for classification. The selection algorithm does not use any information related to the learning algorithm.

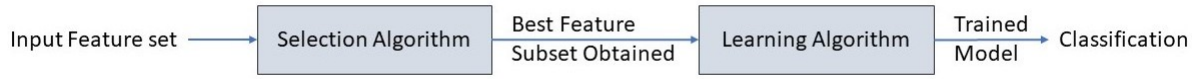


Figure 2.1: Flow Diagram for Filter Approach

Wrapper Approach

In this family of methods, the selection algorithm generates the best feature subset using the learning algorithm. It selects a subset of features and then feeds it to the learning algorithm; the learning algorithm will then execute and provide a feedback to the selection algorithm. This feedback can be any measure used for assessing the model such as accuracy. This feature subset-generation feedback cycle continues until we obtain a feature subset that yields the desired level of performance. The wrapper approach suggests the features which are "most important" for classification.

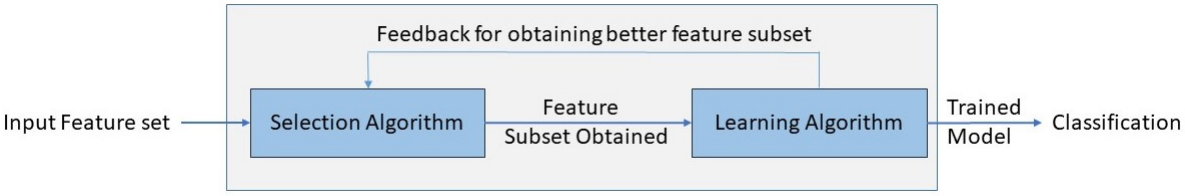


Figure 2.2: Flow Diagram for Wrapper Approach

Embedded Approach

The embedded approach combines the methodology of both filter and wrapper approach. The feature subset from the selection algorithm is used by the learning algorithm for classification and also it concurrently gives the feedback to the selection algorithm. Hence, both the process of selecting the best feature subset and supervised learning are carried out simultaneously. Ñ

2.1.3 Discussion

The choice of the feature selection technique depends on the requirements of the project. In our research project, we need to reduce the feature set size not only for the classifier but also for the interest of the user. The interest of the microbiology side is to make as few measurements (which become data features) as possible. This is because the measurement of a feature costs both money and time for the microbiologists. A research study reveals that the cost of detecting

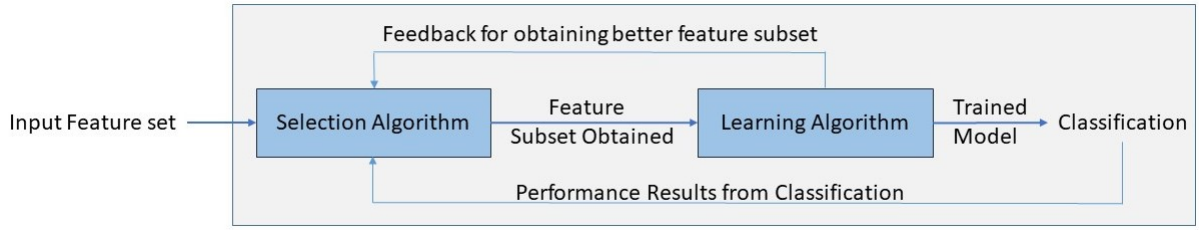


Figure 2.3: Flow Diagram for Embedded Approach

a microbial marker in a water sample costs range from USD 0.6 to USD 5.00 and to obtain the quantitative information of that marker, the cost ranges from USD 0.5 to USD 7.50 [23]. Considering the worst case scenario, if the scientists would incur USD 37.5 for recording the numerical value of an indicator per sample, this is a very costly process. Hence, selecting the smallest best feature subset for classification becomes a prime concern while building the ML model. This would not only improve the model's performance but also help reduce the unnecessary costs incurred by the microbiological scientists.

For these reasons, we do not choose a filter method such as Principal Component Analysis as it looks for a linear combination of the all variables. Therefore, the principal components obtained are linear combinations of the original variables. So, this technique only reduces the number of features for the classifiers but not for the users. The Least Absolute Shrinkage and Selection Operator regression technique is an embedded method of feature selection principles. But the backdrop is that it is a linear method. Since the primary aim of the project is to work on non-linear methods, we had to drop this technique.

RFE approach considers all feature subsets and suggests the best feature subset for classification. This not only promises us good performance but also tells the microbiologists the main features required for good classification. We also studied the limitations of this method, which are listed below:

- If the number of features is very high, then the computational cost is very expensive.
- This method may lead to the problem of overfitting if there is an inadequate amount of observations.

We do accept the limitations of this approach. But, fortunately, the data set used seems to overcome these drawbacks. The data consists of around 10,000 records which is sufficient enough to avoid overfitting. The number of features present are 45 which is not very high. Hence, the RFE approach was chosen for finding the best feature subset. In Chapter 4, we discuss in detail about this approach and also the time taken for executing the algorithm, which is within the acceptable limits.

2.2 Non-Linear Classifiers

Belanche *et al*., focused on Linear methods and artificial neural networks to perform the MST. In this research project, we explore and apply widely known non-linear methods such as Support Vector Machines (with radial kernel) and Random Forests. Our aim is to obtain and compare the performance results and select the best classifier with the best feature subset for classification.

2.2.1 Support Vector Machines

The Support Vector Machine (SVM), is a supervised ML model used for classification. In mathematics, a hyperplane is defined as the plane of *N-1* dimensions formed in a space having *N* dimensions. In this classification problem, the decision boundary is the hyper plane which separates the two classes. If the data has *N* variables, meaning *N* dimensions, then the entire data is plotted in an *N*-dimension space and the hyperplane, having *N-1* dimensions, separates the data into halfspaces, each part representing a class in the target variable.

The position of the hyperplane depends on how the data is arranged in the space. For the sake of understanding, we consider a data set consisting of two dimensions and two classes as portrayed in Figure 2.4. The two dimensions are named x_1 and x_2 and the data is represented in circles. There are two classes, namely, A and B, and they are represented in green and blue colors respectively. Since the data is two dimensional, the hyper plane, i.e., the decision boundary, is a line that separates the data into these two classes. As you see in figure 2.4, we can make out different decision boundaries that divides the data. Intuitively, an optimal hyper plane would be the one that separates the data to the maximum extent. So, how does the algorithm chooses the optimum one? Firstly, in each class, the algorithm finds out the outer most boundary where the points which are closest to the other class lie on this boundary. In fact, the plane is formed with these points and since these points are helping to form the plane, they are called "Support Vectors". In the figure, the support vectors are encircled in red color. The distance between these two planes is called the "Margin" and the optimal hyper plane would be the one that is equidistant from all the support vectors. Hence, the hyperplane is the plane parallel to and equidistant from the planes formed by the support vectors and is represented in bold and black. The planes which are in purple color are also hyperplanes that correctly separates the two classes but they are not the optimal ones.

The equation of a hyperplane is given by:

$$a_1x_1 + a_2x_2 + a_3x_3 + ... + a_nx_n = b \tag{2.1}$$

Where, a_i is the coefficients in which at least one of them is a non-zero value and b is a constant (since the plane is not passing through origin in this case) [24]. Further, equation 2.1

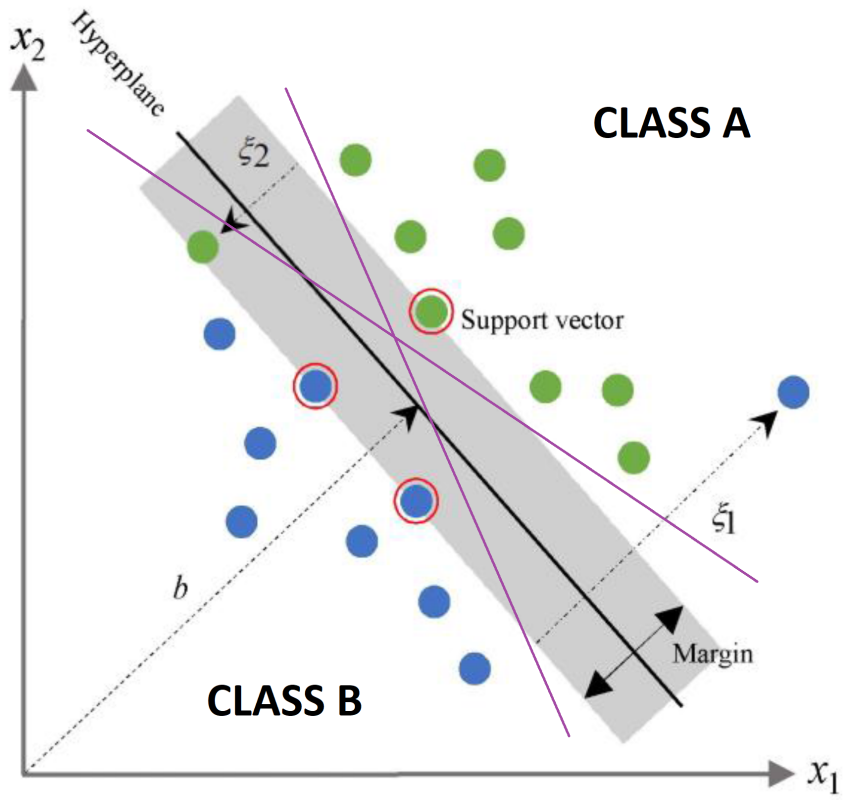


Figure 2.4: Schematic Representation of SVM

can be re-written as:

$$\sum_{i=1}^{n} a_i x_i = b \tag{2.2}$$

$$\implies \sum_{i=1}^{n} a_i x_i - b = 0 \tag{2.3}$$

In the context of SVM-based feature selection, we interpret the coefficients as weights. Therefore every feature has a weight associated with it. And b is defined as the bias which shifts the SVM away from the origin.

Therefore, in the vector form, the equation of hyperplane is given by:

$$\vec{w} \cdot \vec{x} - b = 0 \tag{2.4}$$

The equation of the hyperplanes that bounds the upper and lower parts of the margin is given below respectively.

$$\vec{w} \cdot \vec{x} - b = 1 \tag{2.5}$$

and

$$\vec{w} \cdot \vec{x} - b = -1 \tag{2.6}$$

According to figure 2.4, let us assume class A to have the label as 1 and that of class B as -1. If y_i is the label of the observation x_i, then, we express the SVM as:

$$y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 \qquad \forall i \in 1, ..., n \tag{2.7}$$

Given a test data x, the SVM function is defined below. This function essentially tells in which class the sample x lies.

$$SVM(x) = sgn(\sum_{i=1}^{n} w_i(x \cdot x_i) - b) \tag{2.8}$$

From equation, 2.5 and 2.6, we see that the distance between the parallel lines is 2/||w||. The aim is to maximize this margin region to achieve better classification. Therefore, we need to maximize the term 2/||w||, which implies minimizing ||w||.
With all this information, the formal objective function becomes:

Minimize ||w|| subjected to the condition,

$$y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 \qquad \forall i \in (1, n) \tag{2.9}$$

In Figure 2.4, we observe that there is a misclassification of two data points. A green point is in class B and a blue point is in class A. This is because SVMs allow some amount of misclassification to happen. The extent of this allowance is given by slack variables ξ. The points which fall inside margin (i.e. misclassified) are termed *bounded support vectors* and the points that lie on the boundary of the class are called *unbounded support vectors* [25].

So far, we have discussed the SVM in the linear context, i.e., linear SVM. If the data is not linearly separable, then we need to use non-linear techniques. The motivation for non-linear approach is that we transform the data plotted in the given space into a new higher dimensional space. This new space is formally a Hilbert space and data transformed into this space has higher chances of being quasi-linearly separated. The decision boundary becomes non-linear when it is transformed back to the input space. The transformation of the data from the normal input space to the Hilbert space can be done using a kernel. The kernel function corresponds to an inner product in the Hilbert space. The relationship between the kernel and the feature map is is given by equation 2.9.

$$k(x, z) = < \phi(x), \phi(z) >_H \tag{2.10}$$

Where $< ., . >_H$ denotes inner product in the Hilbert space. The optimal hyperplane is formed by computing the inner products from the Hilbert space. This is called the kernel trick

and there are different kernels to implement this trick. These include radial basis, polynomial and sigmoid functions. We use the radial kernel, since we are dealing with a multi-class classification using non-linear method and also the radial kernel does not consider any prior information of the data.

So far, we have discussed about classification of two classes. This technique can be extended to multi-class classification as well. There are two approaches to do this: One-Vs-All (OVA) and One-Vs-One (OVO).

One-Vs-All

In this approach, if there are *k* classes, then there will *k* SVMs, built where in each SVM, one of the *k* classes formed will be separated from rest of the classes. In every binary classifier formed, the two classes are: one of the *k* classes and the remaining *k-1* classes all put together represents the other class. The confidence score is obtained from each classifier for the decision that it makes. The Label, corresponding to the classifier that produced the highest confidence value is the final class label predicted. This approach is relatively faster.

One-Vs-One

Let us assume that the data set used for classification has *k* class labels in the target variable. This approach builds multiple binary classifiers, where every SVM formed will separate one of *k* classes from one of the remaining *k-1* classes. It essentially builds an SVM for all possible combinations of the class labels. Going by this logic, there will be *k**(*k*-1)/2 SVMs built. The test sample will be classified with each of these SVMs and notes down the class labels obtained in each class. The label classified by majority SVMs is the final label assigned. This approach performs better as it breaks down the problem into "atomic" pieces by constructing all possible SVMs and makes will not create a issue if the data is unbalanced. The drawback is it is computationally expensive.

Discussion

For our project, we have 4 class labels. We would require to build 4 SVMs if we adopt the OVA and 4*(4-1)/2 = 6 SVMs in the case of OVA. Since there is not much difference in the number of SVMs produced, the difference in the computational costs is almost the same. Hence, we chose the OVO approach and we use the LIBSVM as this library implements the SVM in OVO approach.

2.2.2 Random Forests

Before diving into this topic, we shall refresh a few basics of Decision Trees.

Decision Tree

A Decision Tree is a supervised ML model. It is intuitive and easy to understand. The model makes use of a tree structure for modeling. The root and the internal nodes are the decision nodes using the variables of the data. The leaf nodes are the final class label classified for the given sample. The given sample starts its way from the root node and takes the path depending of the decisions it make in the root and internal nodes. Once it reaches the leaf node, the label of the leaf node is the final predicted class for that sample.

Now a question arises. Which variables has to be placed in which decision nodes? If we find a variable that perfectly classifies into distinct classes, then that variable is placed in the root node and the resultant children nodes become leaf nodes. If all the existing variables do not classify into distinct categories, then the leaf nodes are said to be *impure* as misclassification exist. We need to choose the best variables to can provide the minimum misclassifications. Gini impurity measures how frequently a sample can get misclassified. Therefore, lesser is the Gini impurity of the split on a variable, more is preference of that variable and will be placed in higher level nodes. The Gini impurity for a split on a particular variable is given by the formula:

$$\boldsymbol{G} = 1 - \sum_{i=1}^{K} {p_i}^2 \tag{2.11}$$

Where $\boldsymbol{G}$ is the Gini impurity, K is the number of classes and p_i is the probability of an item in the i^{th} class [26].

From Decision Tree to Random Forests

A Random Forests (RF) is an ensemble of decision trees. The motivation behind the idea is that a learning made from a group of individuals would provide better results and enables the knowledge to share with each other. A knowledge gained from a group of individuals would be more effective than from an individual alone. Each decision tree predicts and outcome RF collects the responses obtained from each tree and makes the overall decision. For classification, the class label classified by majority of the decision trees is the final predicted class given by the RF. In the case of regressions, the output of the RF is the mean of the outcomes produced by all the trees in the forest.

In Figure 2.7, a sample $\mathbf{X}$ is given to each of the decision trees and the outcomes of each tree are then collected and k the final prediction of the RF.

Bootstrap Aggregating (or bagging) is the process of splitting the samples randomly with replacement into B bags. Each bag is used to construct a decision tree and therefore B decision trees will be constructed. In theory, one can show that when bootstrapping the data, only two-thirds of the data is allotted to a tree. The remaining one-third of data is termed as Out-Of-Bag (OOB). The advantage of random forest is that it makes use of this remaining one-third of data

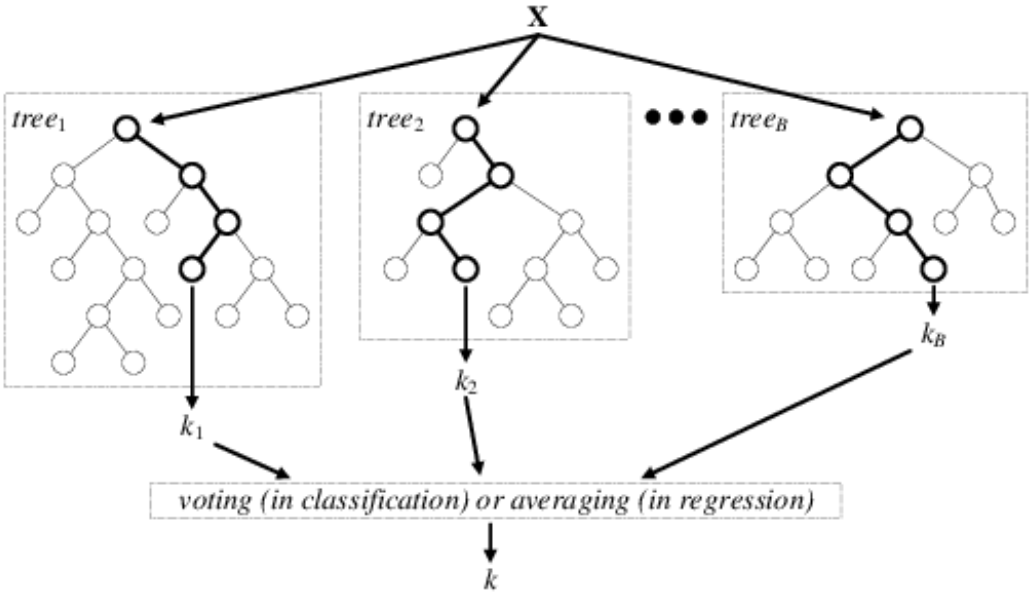


Figure 2.5: Schematic Representation of a Random Forest

to find the error, which is known as Out-Of-Bag error estimate [27].

If there are p predictors, it is generally recommended to have $\sqrt{p}$ feature candidates for each split in the case of a classification problem and $p/3$ feature candidates for that in the case of a regression problem.

What makes the RF random is that it selects a random subset of features for constructing a decision tree in the forest. Since the features selected and the data bootstrapped is also random, this increases the diversity in the forest making it powerful and improves the performance of the classification. Because of this randomness, it avoids the problem of overfitting which is an advantage of this technique. One more benefit of this model is that it computes the importance of each feature and prefers the variables which are more important for classification. RF become computationally expensive if the number of trees is very high.

Chapter 3

Materials: Environmental Data

In this chapter, we provide an overview and insights of the data used for the analyses, as well as some details of their pre-processing to obtain a "clean" set for analysis.

3.1 Exploring the Data

The primary data is collected by microbiologists. Each observation recorded in the data consists of the measurements of microbial and chemical markers of the contaminated water sample. This is a "raw" data matrix formed by making real measurements from real water samples. But there is catch here. The time elapsed when the observation has been taken after the water got contaminated is unknown. Due to this, there will be changes in the values of the indicators due to degradation of the sample. This problem is called *aging*. Water contamination can occur in the sea water and sometimes water bodies can also have high percentage of salinity [28]. Due to the concentration of salinity in water, dilution occurs in the contaminated water. Both aging and dilution problems can lead to the incorrect training of the learning model and ultimately produce erroneous results. Hence, these problems should be solved by simulation before performing the data analysis. Thankfully, the data received for the project has already been simulated to account for and discount the aging and dilution problems. This simulated data is now considered for the data analysis.

As mentioned before, the data comprises of 10,000 observations and has 47 variables, out of which 45 are chemical and microbial indicators. The first 30 predictors include markers of human viruses, host-specific bacteria, host mitochondrial DNA, host-specific bacteriophages and artificial sweeteners such as saccharin. Standard microbial markers used for assessing the load of fecal contamination are also included. There are two target variables, mentioned in the table below. The next 15 variables are the ratios calculated from the indicators (viz. the first 30 variables). The values given in the data set is either 0, which indicates the absence of the indicator, or $log10$ of the original concentration.

Table 3.1: Predictors of the considered Data set

S.No.	Name of the Predictor	Description
1	EC	Enumeration of Escherichia Coli
2	FE	Enumeration of Fecal Enterococci
3	CP	Enumeration of Clostridium Perfringens Spores
4	SomPhg	Enumeration of Somatic Coliphages
5	HMBactPhg	Enumeration of Human specific Bacteroides phages
6	CWBactPhg	Enumeration of Cow specific Bacteroides phages
7	PGBactPhg	Enumeration of Pig specific Bacteroides phages
8	PLBactPhg	Enumeration of Poultry specific Bacteroides phages
9	BifSorb	Enumeration of Human Bifidobacterium Sorbitol Agar
10	BifTot	Enumeration of Human Bifidobacterium Sorbitol Agar
11	HMBif	qPCR Human specific Bifidobacteria
12	CWBif	qPCR Cow specific Bifidobacteria
13	PGNeo	qPCR Pig specific Neoscardovia
14	PLBif	qPCR Poultry specific Bifidobacteria
15	TLBif	qPCR Total Bifidobacteria
16	NoV	qPCR Norovirus
17	BacR	qPCR Ruminant specific Bacteroidetes
18	Pig2Bac	qPCR Pig specific Bacteroidetes
19	AllBac	qPCR All Bacteroidetes
20	HF183TaqMan	qPCR Human specific Bacteroidetes
21	FEqPCR	qPCR Fecal enterococci
22	HMMit	qPCR Human specific Mithocondrial marker
23	CWMit	qPCR Cow specific Mithocondrial marker
24	PGMit	qPCR Pig specific Mithocondrial marker

Continued on next page

Table 3.1 – continued from previous page

S.No.	Name of the Predictor	Description
25	PLMit	qPCR Poultry specific Mithocondrial market
26	Adeno	qPCR Human specific Adenovirus
27	Acesulfame	Artificial sweetener
28	Cyclamate	Artificial sweetener
29	Saccharain	Artificial sweetener
30	Sucralose	Artificial sweetener
31	CLASS	Target variable. The class labels are human and non-human
32	TARGETtype	Target variable. The class labels are HM, CW, PL, PG representing human, cow, poultry and pig respectively
33	SomPhg.HMBactPhg	Ratio of enumeration of somatic coliphages to human specific Bacteroides phages
34	SomPhg.CWBactPhg	Ratio of enumeration of somatic coliphages to cow specific Bacteroides phages
35	SomPhg.PGBactPhg	Ratio of enumeration of somatic coliphages to pig specific Bacteroides phages
36	SomPhg.PLBactPhg	Ratio of enumeration of somatic coliphages to poultry specific Bacteroides phages
37	BifTot.BifSorb	Ratio of enumeration of cultivated total bifidobacteria to sorbitol fermenting bifidobacteria
38	TLBif.HMBif	Ratio of qPCR enumeration of total bifidobacteria to human associated bifidobacteria
39	TLBif.CWBif	Ratio of qPCR enumeration of total bifidobacteria to cow associated bifidobacteria

Continued on next page

Table 3.1 – continued from previous page

S.No.	Name of the Predictor	Description
40	TLBif.PGNeo	Ratio of qPCR enumeration of total bifidobacteria to human associated Neoscardovia
41	TLBif.PLBif	Ratio of qPCR enumeration of total bifidobacteria to poultry associated bifidobacteria
42	AllBac.BacR	Ratio of qPCR enumeration of total Bacteroidales to ruminant associated Bacteroidales
43	AllBac.Pig2Bac	Ratio of qPCR enumeration of total Bacteroidales to pig associated Bacteroidales
44	AllBac.HF183Taqman	Ratio of qPCR enumeration of total Bacteroidales to human associated Bacteroidales
45	FeqPCR.BacR	Ratio of qPCR enumeration of fecal enterococci to ruminant associated Bacteroidales
46	FeqPCR.Pig2Bac	Ratio of qPCR enumeration of fecal enterococci to pig associated Bacteroidales
47	FeqPCR.HF183TaqMan	Ratio of qPCR enumeration of fecal enterococci to human associated Bacteroidales

3.2 Box plots of the data

In statistics, a box plot (or box-and-whisker plot) is a graphical representation of data depicting its quartiles. It consists of a box in which the middle line represents the median and the edges represents the first and third quartile. Two long lines (also known as whiskers) are sprouted from the edges of the box and they extend no more than 1.5 times the interquartile range from the edges. The points lying beyond 1.5 times the interquartile range from the edges are considered univariate outliers [29].

Box plots of each indicator and ratio in our data set are constructed (Since the target variables are categorical, they are not considered for building the box plot). Box plots representing a couple of features are displayed for illustration in figures 3.1 and 3.2.

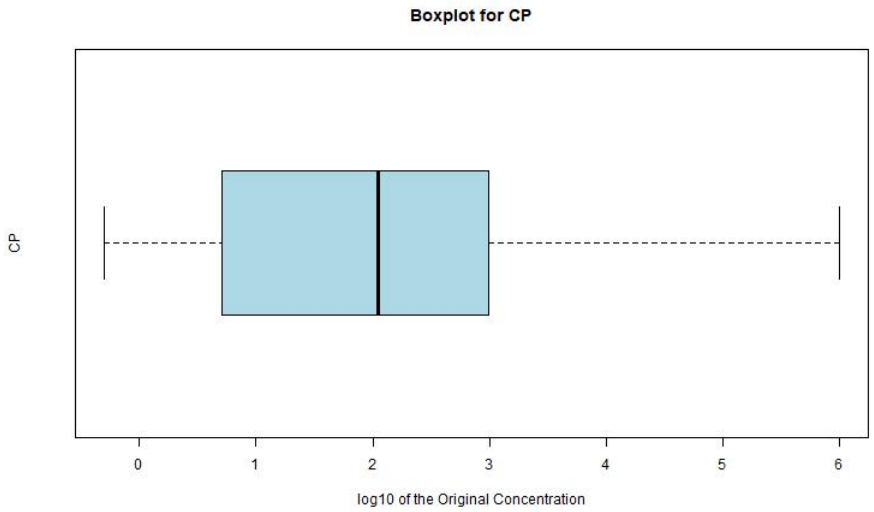


Figure 3.1: Box plot of Clostridium Perfringens spores (CP) indicator

An R script was written and executed to produce the box plots of all the required variables. From the box plots, some insights are obtained and noted. These are listed below:

1. Some variables do not have a single outlier at all. These are features with serial numbers: 3,9,10,15,16,19,22,33,35,36, and 38 to 44.

2. A few variables had very few outliers (a maximum of 4), but even these do not fall very far away (i.e . not much greater than 1.5 times IQR) and they are near the whisker. These are features with serial numbers: 4 and 34.

3. A few variables show many outliers, but they fall close to the whisker. These are features with serial numbers: 1 and 2.

4. Some variables have many outliers falling very far from the whisker. In some cases, the box itself is not even formed (indicating that quartiles and the median coincide). These are features with serial numbers: 5 to 8, 11 to 14, 17, 18, 20, 21, 23 to 30, 37, and 45 to 47.

3.3 Data Cleaning

It is now essential to find out records which have zeroes in all the variables. We check the presence of zeroes only in the indicators as the ratios are calculated from these indicators only. Using the `rowSums()` in R, we found out that 56 such records existed. It does not make sense to have a record with all zeroes and having some random class label assigned to it as this would incorrectly train the model consequently leading to wrong results. Therefore, we removed them permanently from the data set. The algorithm used to detect and remove such records is outlined

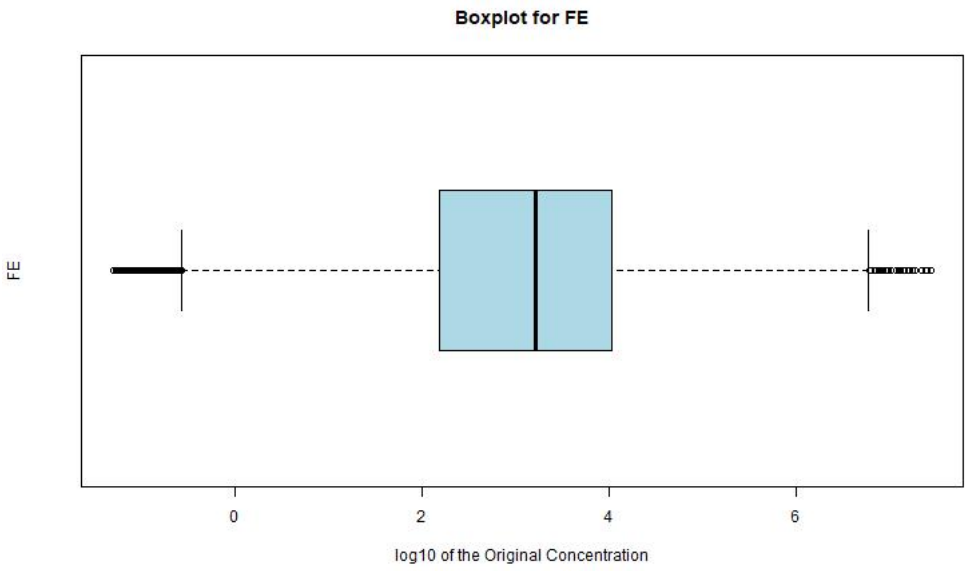


Figure 3.2: Box plot of Fecal Enterococci (FE) indicator

below:

Algorithm 1: Detect and remove records with all zero values in indicators

Result: A cleaned data set

Import the considered data set
Store the indicators alone in *singles*
rowsums := compute the row sum of each record in *singles*
for *each record i in singles* **do**
    if $rowsums_i$ *is not 0* **then**
        $indices_i$:= *TRUE*
    else
        $indices_i$:= *FALSE*
    end
end
newset := records whose *indices* value is *TRUE*
Print the number of records with all zero values
Export *newset*

Chapter 4

Experiments: Feature Selection in classification

In this chapter, the details of the use of RFE algorithms for SVM and RF are presented and discussed. The presentation is carried out in two sections corresponding to the two techniques used. In each section, the feature rank lists produced and the feature subsets selected, for every setting, are elaborated.

4.1 Support Vector Machine - Recursive Feature Elimination

This wrapper technique was first proposed by Guyon *et al.* [30]. It makes use of the SVM model to obtain the weights of each feature in the feature set. The model uses Gaussian kernel and an optimal value for gamma parameter is required for the kernel. The optimum value is estimated from the data set. The `sigest()` in R takes a portion of the data set and estimates an optimal gamma parameter for the Gaussian kernel [31]. Using this parameter, the model is trained with the training data and the weights are obtained from the trained model. The squared weights of each feature are then used to rank the features in decreasing order of relevance. The least relevant feature is taken out and is placed at the bottom of the final rank list, thus, filling the rank list in a bottom-up fashion. Although it is easy to obtain the weights for a binary class scenario, it becomes more complicated when obtaining the weights in a multi-class scenario. In this case, the weights are obtained from every SVM model formed through an OVO approach.

Algorithm 2: Support Vector Machine - Recursive Feature Elimination

Result: A rank list of features with the most relevant features, for classification, on the top.

```
Import the considered data set
currentFeatureSet := All features present in the data set
while currentFeatureSet is not empty do
    Get gamma estimation from the data set for radial kernel for its use in SVM
    Train the SVM Model using radial kernel
    Get the weights from the trained model
    weights := weights*weights
    Create a list by sorting the features based on weights in ascending order
    lowScoredFeature := first element of the list
    Put lowScoredFeature at the bottom of the rankList
    Remove lowScoredFeature from the currentFeatureSet
end
Print rankList
```

4.2 Feature Subset Performance using SVM

From the SVM-RFE, a rank list of the feature candidates is obtained. The next task is to analyze the performance of each feature subset and choose the subsets that are interesting. For each feature subset formed, we train and validate the SVM. The two-fold cross-validation approach is applied to validate the SVM. Because of the relatively big size of the data set, a two-fold cross-validation approach is enough. After the cross-validation, the confusion matrix is obtained using the validation data. The Matthews Correlation Coefficient is computed from the confusion matrix. The mean of MCC is computed from the confusion matrices obtained during cross-validation. The cost corresponding to the best MCC is chosen for training the SVM using the combined data of training and validation sets. The final model obtained is used to predict the test set and the confusion matrix is constructed. From this matrix, the final MCC is computed for this feature subset. The last element of the rank list is removed and the entire process is repeated for the new subset of features. The algorithm terminates when features in the rank list are exhausted.

A line graph is plotted with every feature subset size against its corresponding MCC. Inferences are drawn from this graph and by balancing the feature subset size with the performance; interesting feature subsets are selected for each setting. The algorithm to carry out this process is provided below:

Algorithm 3: Computing the MCC, for every feature subset, from predictions of SVM

Result: A plot depicting the MCC for every feature subset size.

1 *rankList* := import the final list of features obtained from SVM - RFE
2 Import the considered data set
3 Split the data set into *trainingSet*, *validationSet* and *testSet* equally.
4 **while** *rankList is not empty* **do**
5 Update the *trainingSet*, *validationSet* and *testSet*, with the variables that are currently present in the *rankList*
6 *costValues* := a vector of cost values
7 **for** *every* C_i *in costValues* **do**
8 Get *gamma* estimation from the *trainingSet* for radial kernel for its use in SVM
9 Using the *trainingSet*, Train the SVM with *gamma* and C_i as parameters
10 Predict the *validationSet* and get the confusion matrix
11 Compute MCC_1 from confusion matrix
12 Swap *trainingSet* and *validationSet*
13 Repeat steps 8, 9, 10 and 11 to get MCC_2
14 Compute arithmetic mean of MCC_1 and MCC_2 and store it in *MCCForEveryCost*
15 **end**
16 *bestCost* := The value in *costValues* coresponding to the Maximum MCC value in *MCCForEveryCost*
17 *combinedSets* := *trainingSet* $\cup$ *validationSet*
18 Get *gamma** estimation from the *combinedSets* for radial kernel for its use in SVM
19 Using the *combinedSets*, Train the SVM with *gamma** and *bestCost* as parameters
20 Predict the *testSet* and construct the final confusion matrix
21 Compute the MCC from the final confusion matrix and store in *MCClist*
22 Remove the last element in the *rankList*
23 **end**
24 Plot a line graph with MCC values in *MCClist* against its corresponding feature subset size

4.3 Random Forests - Recursive Feature Elimination

In this technique, the wrapper approach uses the RF algorithm as the learning algorithm. The advantage of RF is that cross-validation is not required as we can make use of the OOB estimate to get the parameter of the model. Firstly, the optimal number of trees to construct is obtained by training the RF model and choosing the number of trees that correspond to the lowest OOB error value. With the obtained optimal number of trees, we train the RF. As discussed in the section 2.2.2, $\sqrt{p}$ number of predictors are tried for each split, where p is the number of predic-

tors. From the trained model, we get the Mean Decrease in Gini Index and this metric tells how much decrease is there in Gini Index when a split of node is made on a particular variable. The bigger the decrease, the more important that variable is. Based on the mean decrease in Gini index, the variables of the current feature set are then sorted and the least important variable is taken out and is placed at the bottom of the rank list, filling up the rank list in bottom-up approach again. The algorithm is provided below:

Algorithm 4: Random Forest - Recursive Feature Elimination

Result: A rank list of features with the most relevant features, for classification, on the top.

```
Import the considered data set
currentFeatureSet := All features present in the data set
while currentFeatureSet is not empty do
    numTrees := a vector of possible number of trees to build the RF
    for t_i in numTrees do
        Train the RF using t_i
        Get the error value for this model
    end
    optimumTrees := the value in numTrees corresponding to least error value
    mtry := sqrt(size(currentFeatureSet))
    Train the RF using the optimumTrees and mtry
    Get the importance from the model
    Create a list based on the obtained importance in ascending order
    lowScoredFeature := first element of the list
    Put lowScoredFeature at the bottom of the rankList
    Remove lowScoredFeature from the currentFeatureSet
end
Print rankList
```

4.4 Feature Subset Performance using Random Forests

Similar to that of SVM, the rank list obtained from Random Forests - Recursive Feature Elimination (RF-RFE) is used to generate the feature subset and the performance is evaluated using the MCC. The algorithm for generating the plot is provided below:

Algorithm 5: Computing the MCC, for every feature subset, from predictions of RF

Result: A plot depicting the MCC for every feature subset size.

```
rankList := import the final list of features obtained from RF - RFE
Import the considered data set
Split the data set into trainingSet and testSet in 2:1 ratio
while rankList is not empty do
    Update the trainingSet and testSet, with the variables that are currently present in the
     rankList
    numTrees := a vector of possible number of trees to build the RF
    for t_i in numTrees do
        Train the RF using t_i
        Get the error value for this model
    end
    optimumTrees := the value in numTrees corresponding to least error value
    mtry := sqrt(size(currentFeatureSet))
    Train the RF using the optimumTrees and mtry
    Predict the testSet using the trained model and construct the confusion matrix
    Compute MCC from the confusion matrix and store it MCClist
    Remove the last element in the rankList
end
Plot a line graph with MCC values in MCClist against its corresponding feature subset
 size
```

4.5 Graphs and Discussion

Before discussing the results reported in the graphs, we first discuss the execution time of the algorithms. The speed of execution of both (SVM and RF) the selection techniques are assessed. The total time taken for executing the RFE and plot generation algorithms of SVM type and RF type are summarized below:

	Single and derived markers	**Single markers only**
Multiclass Classification	2476.14	1199.81
Binary Classification	1170.73	623.41

Table 4.1: Total Execution time(in seconds) for SVM selector

Tables 4.1 and 4.2 show the total times of execution of the selectors. From the tables, it is inferred that the SVM selector executes relatively faster than that of the RF. Further, from both

	Single and derived markers	Single markers only
Multiclass Classification	5697.01	2462.75
Binary Classification	4876.17	2072.04

Table 4.2: Total Execution time(in seconds) for RF selector

the techniques, we see that the multi-class classification comparatively takes more time than the binary classification. The decrease in execution time from the 'single and derived markers' case to 'single markers only' case is obvious as there is a lesser number of features in the latter one.

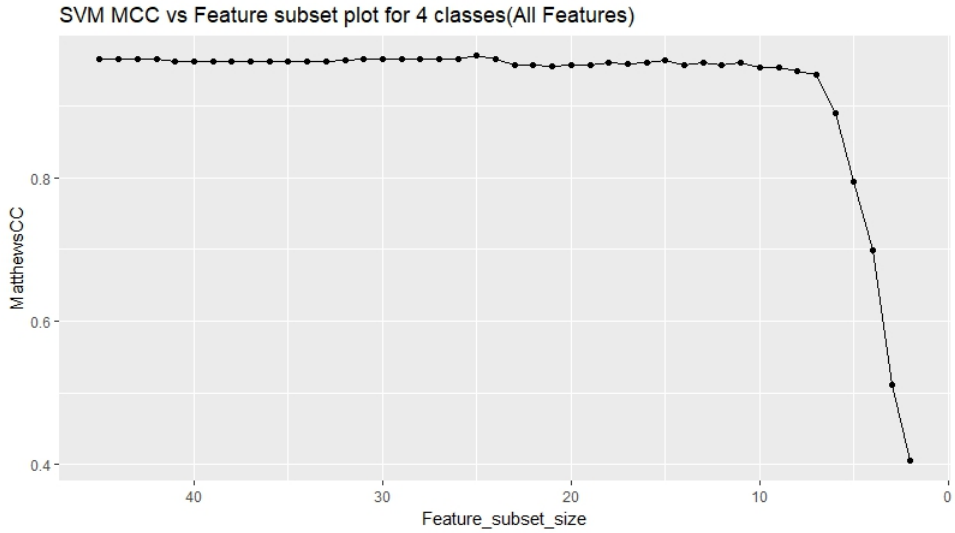
4.5.1 SVM Performance

It is interesting to study the plots by observing how the classifier performs when features are removed one by one from the set. That is why the plot starts from the maximum subset size at the left and size decreases as we go to the right. We look out for the subset size for which there is a sudden decrease in the performance as the decrease indicates the presence of "important" features at that particular subset. The cost and time factors for the microbiologists depends on the features present in the subset. Sometimes, the cost to obtain a feature subset of lesser size may be more than that of a feature subset whose size is comparatively bigger. Thus, in many cases, more than one feature subsets are selected as it is a requirement of the microbiologists. We now discuss the plots depicting the performance of SVM. With this in mind, let us first consider the multi-class classification.

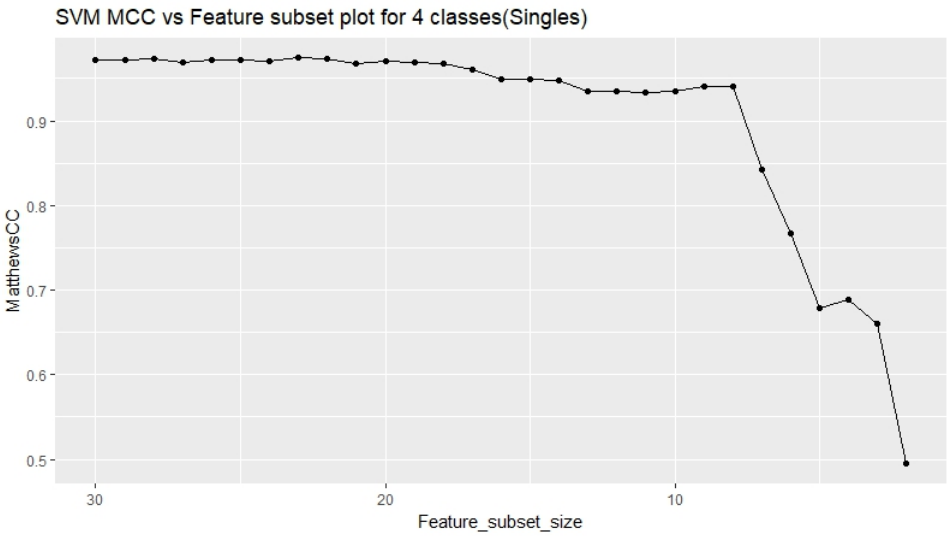
Multiclass Classification

In the single and derived features case, there is almost a straight line, up to 7. This means the performance remains same for most of the larger subsets and has an MCC well over 0.9. This indicates the presence of many redundant features. It is inferred that starting from subset size 45 down to 7, similar performance can be achieved. Therefore, we can eliminate most of the features and consider a subset with size 7. If we further remove features from size 7, there is a sudden drop in performance which indicates that the features of this set are more "important" for classification. Therefore subset size 7 is chosen for this case.

The plot formed using single markers only, seems to drop in the midway. The performance slightly decreases from size 18 but maintains MCC values over 0.9. From subset size 14, performance drops below 0.95 but slowly increases at size 8. From 8, it drastically drops down, indicating that some important features are being removed. In the end, we see a small rise when size is 4. It is inter sting to study how classifier performs with subset size 4. With these indications, we consider some interesting subsets for final classification. These are sizes 4, 8 and 14.



(a) Performance of SVM using single and derived features.



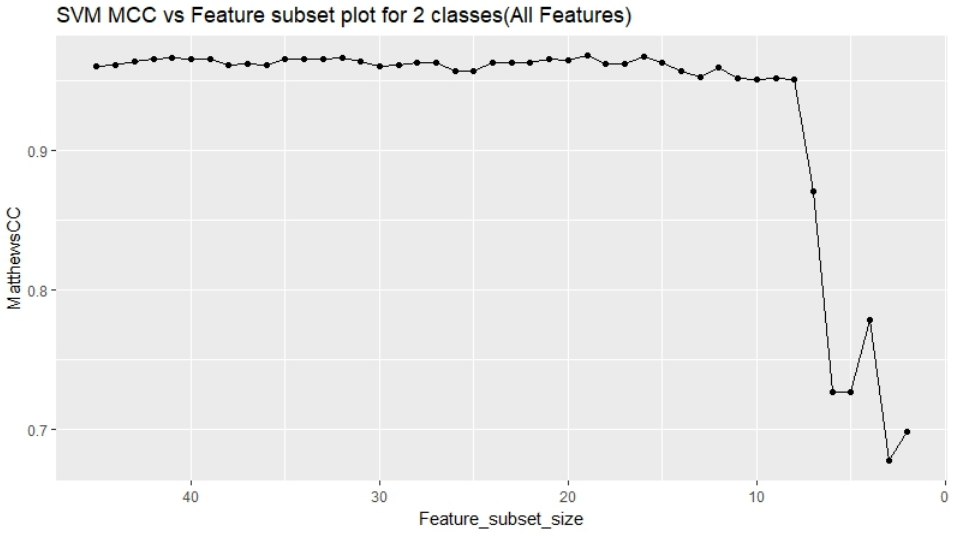
(b) Performance of SVM using single quantities.

Figure 4.1: SVM Performance for multiclass classification.

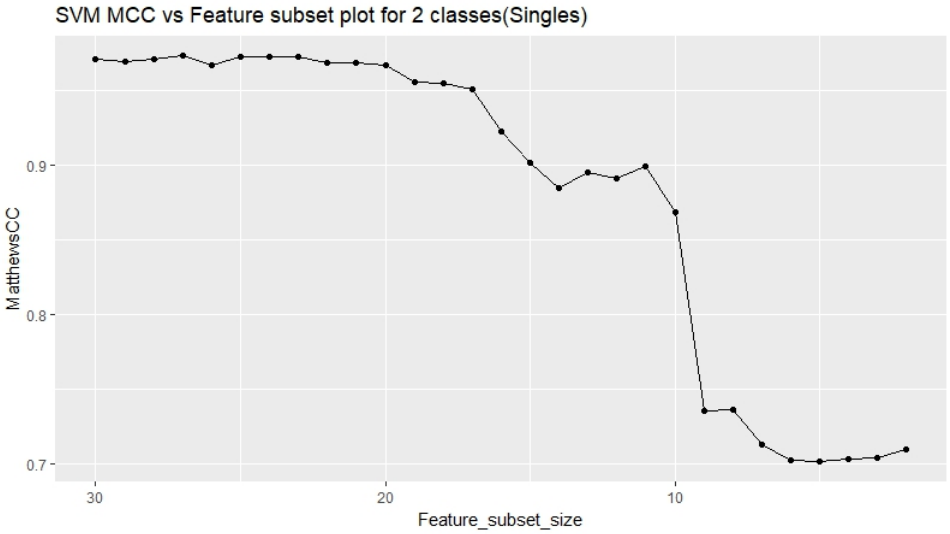
Binary Classification

In the single and derived features case, the performance stays above 0.95 up to size 8, but the line is quite bumpy. There are ups and downs along the path. On a closer look, we find that there is a steady decrease at size 16 although the fall halts after 3 removals. It would be nice to see how size 16 performs in classification. After size 8, it steeply decreases. It is astonishing to see there is a sudden increase at size 4 and it is interesting to see how the classifier performs with size 4. So, subset sizes 16, 8 and 4 are selected.

In the plot formed using single markers only, the performance drops early as compared with the previous cases. From size 17, it falls steadily to below 0.9 MCC but later rises up and



(a) Performance of SVM using single and derived features.



(b) Performance of SVM using single markers only.

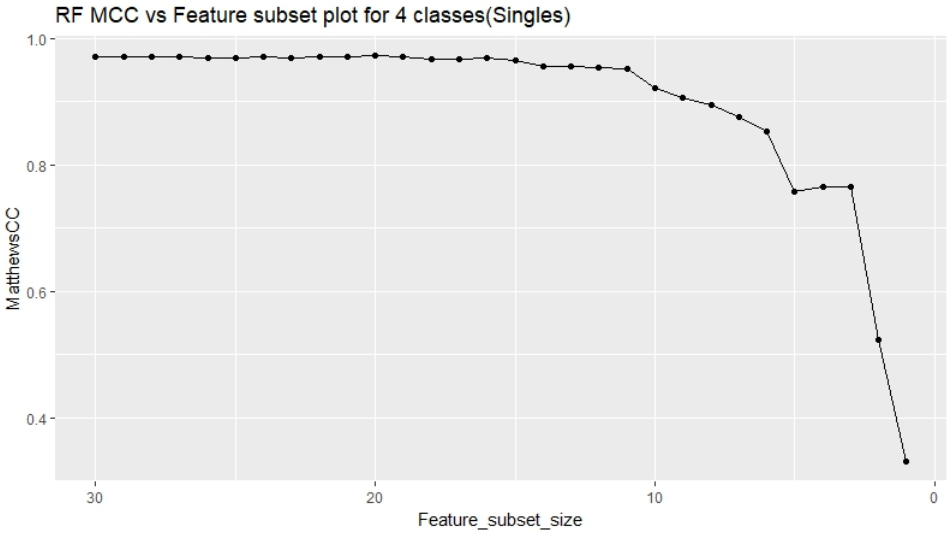
Figure 4.2: SVM Performance for binary classification.

touches 0.9 MCC at size 11. After size 11, the performance drops steeply but suddenly maintains up to size 8 and the decreases. It is worth to select these critical points (feature subsets) in the plot and assess the classifier's performance. Therefore, subset sizes 8, 11 and 17 are selected.

4.5.2 Random Forests Performance

In this subsection, we make inferences and discussions on the RF' performances for all the settings.

(a) Performance of Random Forests using single and derived features.



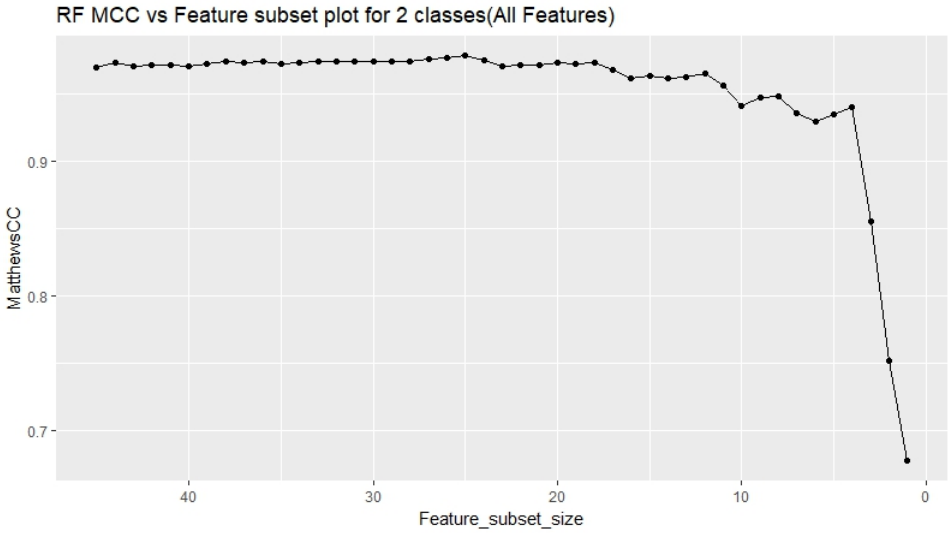
(b) Performance of Random Forests using single markers only.

Figure 4.3: Random Forest performance for multi-class classification.

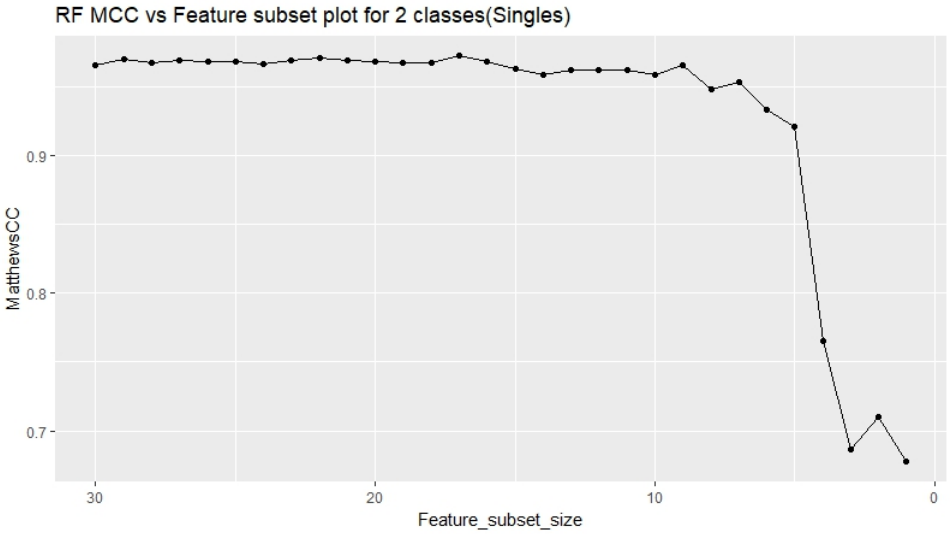
Mutliclass Classification

In the single and derived features case, there is almost a straight line up to 18 features indicating the presence of redundant features. The line faintly drops later. From 17 features, there seems to no considerable fall in the performance and maintains a performance of over 0.9 as measured by MCC. At size 10, there is a considerable decrease in MCC. The fall is steeper at size 7 and drastically dips at size 5. There is a slight increase when subset size is 3, but this increase is not significantly high as compared to that in Figure 4.2(a). Moreover, it displays the characteristic of Randomness. Therefore, we select feature subsets of sizes 5, 7 and 10 in this scenario.

In the plot formed using single markers only, starting from size 30 to 11, there is almost a



(a) Performance of Random Forest using single and derived features.



(b) Performance of Random Forests using single markers only.

Figure 4.4: Random Forests performance for binary classification.

line which maintains well over 0.95 MCC. All these features do not add significant value to the performance and can safely be removed. After removing a feature when subset size is 11, the performance starts to drop slowly and at 6, there is a sudden decrease. Similar to the previous case, the performance is maintained at sizes 5, 4 and 3 but there is no significant rise. It later falls down to below 0.4 MCC which indicates the most relevant features for classification are removed. Thus, in this case, subset sizes 6 and 11 are chosen for classification.

Binary Classification

In the single and derived features case, there is almost a straight line up to size 12. Even though the line is quite bumpy, it manages to maintain a very good performance of 0.95+ MCC. But

after 12, there is a considerable decrease in performance. A similar fall is seen in subset size 8 too. After removing a feature when subset size is 4, there is a substantial fall in performance. This indicates that the most relevant features for classification are being removed. Thus, feature subset sizes 4, 8 and 12 are selected.

The plot formed using single markers only, is similar to that of the single and derived features case and manages to maintain 0.95+ MCC with little ups and downs. In this case, the performance drops after 9 features but maintains above 0.9 MCC. After removing a feature when size is 7, the performance keeps dropping.

4.6 Selected Feature Subsets

The feature subsets selected for each of the cases presented above are summarized next:

	Single and derived markers	**Single markers only**
Multiclass Classification	7	4, 8, 14
Binary Classification	4, 8, 16	8, 11, 17

Table 4.3: Selected feature subset sizes for SVM selector

	Single and derived markers	**Single markers only**
Multiclass Classification	5, 7, 10	6, 11
Binary Classification	4, 8, 12	7, 9

Table 4.4: Selected feature subset sizes for RF selector

We also show the names of the features that are present in each of the selected feature subset. For every feature subset, the microbiologists require the feature names and the number of unique features required to achieve good performance of the classifier. As mentioned earlier, ratios are included when considering single and derived features. The intention to have the ratios in the analysis is that in microbiology, there can be cases where two features, say A and B may not act good predictors but a predictor derived by unifying A and B can become a better predictor. If a ratio feature is present in the subset, then it should be considered as two features instead of one because the ratio can only be obtained when the two features are available. For instance, in Table 4.5, we see that the subset size is 7 and this includes ratios. But the number of different features present in the set is 8. In this case, we see *HMBactPhg* performs well alone and also combined with *SomPhg*. *SomPhg* also performs well with *PGBactPhg* but *PGBactPhg* alone is not present. This shows that *PGBactPhg* solely is not a good predictor but it performs better when combined with *SomPhg*. On the contrary, consider the case of subset size 8 in

Table 4.11, we observe that the number of unique features required is lesser than the size of the feature subset. It is interesting to see that *HMBactPhg* and *SomPhg* are good predictors when considered solely and when both of them are combined. In addition, the feature derived from *SomPhg* and *PGBactPhg* is a good predictor. In most of the other cases, the subset size and the number of different features in the subset remains the same.

Subset size	**Names of the features in the selected subset**	**Number of unique features present**
7	HMBactPhg, SomPhg.PGBactPhg, SomPhg.HMBactPhg, NoV, PLMit, TLBif.PLBif, PGMit	8

Table 4.5: Features for SVM multiclass classification using single and derived features

Subset size	**Names of the features in the selected subset**	**Number of unique features present**
4	HMBactPhg, PGBactPhg, CWMit, Pig2Bac	4
8	HMBactPhg, PGBactPhg, CWMit, Pig2Bac, PLBif, PGMit, PLMit, NoV	8
14	HMBactPhg, PGBactPhg, CWMit, Pig2Bac, PLBif, PGMit, PLMit, NoV, BacR, CWBif, PGNeo, HMMit, HMBif, SomPhg	14

Table 4.6: Features for SVM multiclass classification using single markers only

Subset size	Names of the features in the selected subset	Number of unique features present
4	HMBactPhg, EC, Cyclamate, SomPhg.HMBactPhg	4
8	HMBactPhg, EC, Cyclamate, SomPhg.HMBactPhg, BifSorb, HF183TaqMan, HMMit, NoV	8
16	HMBactPhg, EC, Cyclamate, SomPhg.HMBactPhg, BifSorb, HF183TaqMan, HMMit, NoV, Adeno, HMBif, Acesulfame, CP, PGMit, CWMit, TLBif.CWBif, PLMit	17

Table 4.7: Features for SVM Binary classification using single and derived features

Subset size	Names of the features in the selected subset	Number of unique features present
8	HMBactPhg, Adeno, EC, Pig2Bac, PGMit, Cyclamate, CWBif, CWMit	8
11	HMBactPhg, Adeno, EC, Pig2Bac, PGMit, Cyclamate, CWBif, CWMit, BacR, BifSorb, NoV	11
17	HMBactPhg, Adeno, EC, Pig2Bac, PGMit, Cyclamate, CWBif, CWMit, BacR, BifSorb, NoV, HF183TaqMan, Acesulfame, HMMit, CP, HMBif, PLMit	17

Table 4.8: Features for SVM Binary classification using single markers only

Subset size	Names of the features in the selected subset	Number of unique features present
5	SomPhg.HMBactPhg, SomPhg.PGBactPhg, CP, HMBactPhg, TLBif.PLBif,	6
7	SomPhg.HMBactPhg, SomPhg.PGBactPhg, CP, HMBactPhg, TLBif.PLBif, CWMit, SomPhg.CWBactPhg	8
10	SomPhg.HMBactPhg, SomPhg.PGBactPhg, CP, HMBactPhg, TLBif.PLBif, CWMit, SomPhg.CWBactPhg, EC, NoV, FE	11

Table 4.9: Features for RF multiclass classification using single and derived features

Subset size	Names of the features in the selected subset	Number of unique features present
6	HMBactPhg, PLBif, CWMit, NoV, SomPhg, PGMit	6
11	HMBactPhg, PLBif, CWMit, NoV, SomPhg, PGMit, EC, CP, PGBactPhg, FE, PLMit	11

Table 4.10: Features for RF multiclass classification using single markers only.

Subset size	Names of the features in the selected subset	Number of unique features present
4	HMBactPhg, SomPhg.HMBactPhg, NoV, EC	4
8	HMBactPhg, SomPhg.HMBactPhg, NoV, EC, SomPhg, Cyclamate, SomPhg.PGBactPhg, CP	7
12	HMBactPhg, SomPhg.HMBactPhg, NoV, EC, SomPhg, Cyclamate, SomPhg.PGBactPhg, CP, SomPhg.PLBactPhg, SomPhg.CWBactPhg, FE, TLBif.CWBif	12

Table 4.11: Features for RF binary classification using single and derived features

Subset size	**Names of the features in the selected subset**	**Number of unique features present**
7	HMBactPhg, Cyclamate, EC, NoV, SomPhg, CP, FE	7
9	HMBactPhg, Cyclamate, EC, NoV, SomPhg, CP, FE, BifSorb, TLBif	9

Table 4.12: Features for RF binary classification using single markers only.

Chapter 5

Classification and Results

In this chapter, we use the feature subsets selected in the previous chapter as the basis to train the SVM and RF classifiers. We evaluate the classifier's performance using F1 score and MCC metrics.

5.1 SVM Classification

Now that the feature subsets have already been selected from the SVM selector for each setting, we can perform the classification using these feature subsets and evaluate the performance of the classifier. The evaluation is carried out using MCC and the F1 score for each class. The algorithm for classification using SVM is given below:

Algorithm 6: Assessing SVM's performance for Classification

Result: A confusion matrix, F1 score and MCC evaluating the performance of SVM

1 Import the considered data set
2 *featureSubset* := import the selected feature subset obtained from the SVM selector
3 From the data set, extract those variables that present in the *featureSubset* and store in *data*
4 Split the *data* into *trainingSet* and *testSet* in 2:1 ratio
5 *bestCost* := The cost value used to obtain this feature subset from SVM selector
6 Get *gamma* estimation from *trainingSet* for the radial kernel in SVM
7 Using *trainingSet*, train the SVM with *gamma* and *bestCost* as parameters
8 Predict the *testSet* and construct the confusion matrix
9 Compute overall MCC and class-wise F1 score from the confusion matrix and print them

5.2 Random Forests Classification

From the RF selector, we have obtained the feature subsets for each setting. We now perform the classification using these feature subsets and evaluate their performance using the same met-

rics used for SVM.

Algorithm 7: Assessing RF's performance for classification

Result: A confusion matrix, F1 score and MCC evaluating the performance of RF

1 Import the considered data set
2 *featureSubset* := import the selected feature subset obtained from the RF selector
3 From the data set, extract those variables that present in the *featureSubset* and store in *data*
4 Split *data* into *trainingSet* and *testSet* in 2:1 ratio
5 *optimumTrees* := The optimum number of trees used to obtain this feature subset from RF selector
6 $mtry := \sqrt{size(featureSubset)}$
7 Using *trainingSet*, train the RF with *optimumTrees* and *mtry* as parameters
8 Predict the *testSet* using the trained model and construct the confusion matrix
9 Compute overall MCC and class-wise F1 score from the confusion matrix and print them.

5.3 Metrics used for evaluation

In this section, the metrics used for assessing the model are presented in a tabular form.

5.3.1 Matthews Correlation Coefficient

The MCC is computed directly from the confusion matrix and is one of the most reliable metrics for evaluating the performance of a classifier [32]. The benefit of using MCC is that it captures the entire confusion matrix thus giving a complete insight into the performance. There are seperate formulae for computing the MCC for binary classification and multi-class classification. The former one is represented by equation 5.1 and the latter one by equation 5.2.

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP+FP)(FP+TN)(TN+FN)(FN+TP)}} \tag{5.1}$$

where TP, TN, FP and FN are the True Positives, True Negatives, False Positives and False Negatives in the confusion matrix. This formula is used for confusion matrix obtained from binary classification.

The MCC for the multi-class classification is computed from $K \times K$ confusion matrix C where K is the number of different classes. The generalized formula is given below:

$$MCC = \frac{\sum_k \sum_l \sum_m C_{kk} C_{lm} - C_{kl} C_{mk}}{\sqrt{\sum_k (\sum_l C_{kl})(\sum_{k'|k' \neq k} \sum_{l'} C_{k'l'})} \sqrt{\sum_k (\sum_l C_{lk})(\sum_{k'|k' \neq k} \sum_{l'} C_{l'k'})}} \tag{5.2}$$

In both the cases, the value of MCC lies between -1 and 1. A value of 1 indicates excellent

agreement and a value of -1 indicates very poor prediction. MCC is used to compute the overall performance of the classifier.

5.3.2 F_1 Score

The F_1 score is used to assess a binary classifier. It is mathematically the harmonic mean of precision and recall, which, in terms of the elements of the confusion matrix, is provided below [33].

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{5.3}$$

where *TP*, *FP* and *FN* are the True Positives, False Positives and False Negatives in the confusion matrix. F1 was preferred to accuracy. Consider a confusion matrix with *TP* = 10, *FP*=25 and *FN*=25 and*TN* = 40. Accuracy is computed by taking the ratio of the sum of the diagonal elements to the total number of samples classified. Let us compute the accuracy and F_1 score:

$$Accuracy = \frac{10+40}{10+25+25+40} = 0.5$$

$$F_1 = \frac{2 \times 10}{2 \times 10 + 25 + 25} = 0.28$$

From the expressions above, we observe that F_1 score penalizes more than the accuracy when class labels are imbalanced. Accuracy performs well when the data is balanced but performs poor when it is imbalance. The data we analyzed is not exactly balanced and at the same time not very skewed. All the class labels in the target variable are important and thus, F_1 is chosen for evaluation.

5.4 Results

In this section, the classification results are presented along with the F_1 score and MCC for all the settings. In all the settings, the confusion matrices are first illustrated and following below it are the class-wise F_1 scores and MCC. These are given from Table 5.1 to Table 5.20.

From the tables, we infer that subset size 14 for multi-class SVM using single markers performs comparatively better than other subset sizes of this kind.
In the case of binary SVM using single and derived features, both subset sizes 8 and 16 perform well. Although the MCC for subset size of 16 is slightly above than that of subset size 8, it requires 17 unique features in order to achieve this level. A similar performance can be reached by using 8 unique features only. In contrast, the binary SVM using single markers performs relatively poor with 8 features. But it achieves similar performance when 17 features are taken

which is relatively costly for the microbiologists.

We now turn to the RF technique. Consider the multi-class RF using single and derived features. A subset consisting of 11 unique features performs better than a subset of size 5 and 7. A similar performance, with the same number of unique features, is reached in the case of multi-class RF using single markers only. The RF seem to perform very well for binary classification using single and derived features. Using 4 unique features itself, a high performance of 0.94 MCC is obtained. This is probably the best option for classification as compared to others since a high performance is achieved with lesser features. A similar performance is also achieved from binary RF classifier using single markers only but it needs 7 or 9 unique features to attain the feat.

Overall, we conclude that the binary RF classifier performs well using only 4 single and derived features. Bringing 45 features down to 4 will indeed save a huge amount of time and money for the microbiologists.

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	744	43	39	32
Human	0	900	17	0
Pig	0	2	830	0
Poultry	3	4	9	692
F_1 Score	0.9271028	0.9646302	0.9612044	0.9664804
MCC	0.9409225			
Number of unique features	8			

Table 5.1: Performance of multi-class SVM using single and derived features with subset size 7

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	683	0	0	0
Human	0	629	17	60
Pig	0	1	557	53
Poultry	64	319	338	611
F_1 Score	0.9552448	0.7680098	0.7397078	0.5943580
MCC	0.6888066			
Number of unique features	4			

Table 5.2: Performance of multi-class SVM using single markers only with subset size 4

5.5 Implementation

The execution of the entire set of analyses is carried out on a laptop having specifications of 12 GB RAM and an Intel i3 processor. All the algorithms described in the previous sections are

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	747	46	39	37
Human	0	901	25	0
Pig	0	2	831	1
Poultry	0	0	0	686
F_1 Score	0.9245050	0.9610667	0.9612493	0.9730496
MCC	0.9406719			
Number of unique features	8			

Table 5.3: Performance of multi-class SVM using single markers only with subset size 8

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	733	23	15	20
Human	13	899	9	2
Pig	1	20	854	3
Poultry	0	7	17	699
F_1 Score	0.9531860	0.9604701	0.9633390	0.9661368
MCC	0.947697			
Number of unique features	14			

Table 5.4: Performance of multi-class SVM using single markers only with subset size 14

implemented in R language. The open-source software *RStudio* is used as the GUI for running the R scripts. Separate R scripts were written for all the algorithms covering the four settings using SVM and RFs.

With respect to SVM, the $\{e1071\}$ package is used for training the SVM model. To get the gamma estimation, the `sigest()` from the $\{kernlab\}$ package is used. The $\{randomForest\}$ package is used for applying the RF technique. For both SVM and RF, the $\{caret\}$ package is used for obtaining the confusion matrix and the $\{mccr\}$ package is used to compute the MCC for binary classification. A function has been implemented to compute the MCC for the multi-class classification using equation 5.2. The $\{ggplot2\}$ package is used for generating the plots.

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	680	20
Non-human	269	2346
F_1 Score	0.8247423	0.9419795
MCC	0.7842318	
Number of unique features	4	

Table 5.5: Performance of binary SVM using single and derived features with subset size 4

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	900	27
Non-human	49	2339
F_1 Score	0.9594883	0.9419795
MCC	0.943631	
Number of unique features	8	

Table 5.6: Performance of binary SVM using single and derived features with subset size 8

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	913	15
Non-human	36	2351
F_1 Score	0.9728290	0.9892699
MCC	0.962218	
Number of unique features	17	

Table 5.7: Performance of binary SVM using single and derived features with subset size 16

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	669	66
Non-human	280	2300
F_1 Score	0.7945368	0.9300445
MCC	0.7367369	
Number of unique features	8	

Table 5.8: Performance of binary SVM using single markers only with subset size 8

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	877	66
Non-human	72	2300
F_1 Score	0.9270613	0.9708738
MCC	0.8979443	
Number of unique features	11	

Table 5.9: Performance of binary SVM using single markers only with subset size 11

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	890	8
Non-human	59	2358
F_1 Score	0.9637250	0.0.9859921
MCC	0.9504278	
Number of unique features	17	

Table 5.10: Performance of binary SVM using single markers only with subset size 17

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	704	65	134	97
Human	10	866	14	9
Pig	26	14	742	21
Poultry	7	4	5	597
F_1 Score	0.8059531	0.9372294	0.8739694	0.8930441
MCC	0.8408618			
Number of unique features	6			

Table 5.11: Performance of multi-class RF using single and derived features with subset size 5

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	739	16	76	72
Human	7	903	25	7
Pig	1	26	785	21
Poultry	0	4	9	624
F_1 Score	0.8957576	0.9550502	0.9085648	0.9169728
MCC	0.8954069			
Number of unique features	8			

Table 5.12: Performance of multi-class RF using single and derived features with subset size 7

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	738	15	44	34
Human	8	927	8	0
Pig	1	7	838	22
Poultry	0	0	5	668
F_1 Score	0.9353612	0.9799154	0.9506523	0.9563350
MCC	0.9424506			
Number of unique features	11			

Table 5.13: Performance of multi-class RF using single and derived features with subset size 10

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	747	46	37	157
Human	0	825	1	0
Pig	0	78	841	0
Poultry	0	0	16	567
F_1 Score	0.8615917	0.9295775	0.9272326	0.8676358
MCC	0.8698905			
Number of unique features	6			

Table 5.14: Performance of multi-class RF using single markers only with subset size 6

Confusion Matrix	***Reference***			
Prediction	Cow	Human	Pig	Poultry
Cow	739	40	34	26
Human	8	909	8	0
Pig	0	0	847	0
Poultry	0	0	6	698
F_1 Score	0.9319042	0.9701174	0.9724455	0.9775910
MCC	0.9514609			
Number of unique features	11			

Table 5.15: Performance of multi-class RF using single markers only with subset size 11

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	874	6
Non-human	75	2360
F_1 Score	0.9557135	0.9831285
MCC	0.9401505	
Number of unique features	4	

Table 5.16: Performance of binary RF using single and derived features with subset size 4

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	887	9
Non-human	62	2357
F_1 Score	0.9615176	0.9851620
MCC	0.9474469	
Number of unique features	7	

Table 5.17: Performance of binary RF using single and derived features with subset size 8

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	909	8
Non-human	40	2358
F_1 Score	0.9742765	0.9899244
MCC	0.9644797	
Number of unique features	12	

Table 5.18: Performance of binary RF using single and derived features with subset size 12

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	896	11
Non-human	53	2355
F_1 Score	0.9655172	0.9865941
MCC	0.9525912	
Number of unique features	7	

Table 5.19: Performance of binary RF using single markers only with subset size 7

Confusion Matrix	***Reference***	
Prediction	Human	Non-human
Human	911	11
Non-human	38	2355
F_1 Score	0.9738108	0.9897037
MCC	0.9637122	
Number of unique features	9	

Table 5.20: Performance of binary RF using single markers only with subset size 9

Chapter 6

Conclusion and Future Work

In this final chapter, we draw some brief conclusions from the results presented in the previous chapters. We also summarize the goals achieved and outline a few ideas about potential future work.

6.1 Goals Achieved

At the end of the project, the goals set at the initial stages have been achieved. These are listed below:

- We first explored the microbiological data and applied suitable strategies for their pre-processing and "cleaning" to get them ready for their analysis.
- A few feature selection algorithms for dimensionality reduction were surveyed and the RFE was the final choice for selecting interesting feature subsets.
- The non-linear methods SVM and RF were understood and then applied for creating the data models through the processes of training and classification. Interesting feature subsets were chosen by balancing the subset size with the performance.
- With these interesting feature subsets, the SVM and RF models were trained to predict the test data. Their performance was evaluated using the F_1 score and MCC measures.

6.2 Conclusion

In the brief period of five months, we have carried out a data analysis related to an specific environmental issue. A data set received from five countries of the EU was used for the analyses. Its limited quality required the use of initial pre-processing techniques. Knowledge of R language was gained to carry out data analysis tasks. Using SVM and RF selectors we constructed plots to analyze the model's performance for every feature subset and selected a few feature

subsets of interest. We finished our analyses by proposing feature subsets of interest together with the performances. The outcome of the project will benefit the microbiologists by saving both money and time. Now that the microbiologists have the important feature subsets, they can concentrate on obtaining the features which are much relevant for classification. This will help in determining the source of fecal contamination comparatively quicker and helps in solving the environmental problem.

6.3 Future Work

This project was aimed at improving an aspect of environmental sustainability. While considering the environmental aspects, it would be interesting to study how the season affects the contaminated water. That is, the study of seasonality is worth pursuing. The microbial markers measured from the contaminated water samples can vary depending on when they are measured. Therefore, it will be worthwhile doing another two data analysis in which one uses data taken in the summer season and the other uses the data taken in the winter season.

From an analytical viewpoint, it might be worth exploring different modeling approaches beyond the SVM and RF techniques to discover, more in detail, to what extent the obtained results depend on the choice of model. One could extend this work by applying state-of-the-art techniques such deep learning to look for potential changes specially in terms of the selected feature subsets.

Bibliography

[1] Sara Blumberg. Warm air helped make 2017 ozone hole smallest since 1988, Nov 2017. URL https://www.nasa.gov/feature/goddard/2017/warm-air-helped-make-2017-ozone-hole-smallest-since-1988. Accessed: 25 May 2018.

[2] Alina Bradford. Deforestation: Facts, causes effects, Apr 2018. URL https://www.livescience.com/27692-deforestation.html. Accessed: 25 May 2018.

[3] Spain. URL https://www.indexmundi.com/spain/. Accessed: 5 March 2018.

[4] Lluis Belanche-Munoz and Anicet R. Blanch. Machine learning methods for microbial source tracking. *Environmental Modelling Software*, 23(6):741–750, 2008. doi: 10.1016/j.envsoft.2007.09.013.

[5] Spain population (live). URL http://www.worldometers.info/world-population/spain-population/. Accessed: 15 June 2018.

[6] Livestock population, 2016 (million head). URL http://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Livestock_population,_2016_(million_head).png. Accessed: 30 May 2018.

[7] Spain, a country full of manure. URL https://www.foodandwatereurope.org/blogs/spain-a-country-full-of-manure. Accessed: 30 May 2018.

[8] Agricultural census in spain. URL http://ec.europa.eu/eurostat/statistics-explained/index.php/Agricultural_census_in_Spain. Accessed: 30 May 2018.

[9] Livestock population, 2016 (million head). URL http://ec.europa.eu/eurostat/statistics-explained/index.php?title=Agricultural_production_-_animals. Accessed: 30 May 2018.

[10] Debby Sargeant. Fecal contamination source identification methods in surface water. Technical Report 99-345, Department of Ecology, Washington state, Oct 1999.

[11] Charles Hagedorn, Sandra L. Robinson, Jennifer R. Filtz, Sarah M. Grubbs, Theresa A. Angier, and Raymond B. Reneau Jr. Determining sources of fecal pollution in a rural virginia watershed with antibiotic resistance patterns in fecal streptococci. *Applied and Environmental Microbiology*, 65(12):5522–5531, Dec 1999.

[12] Shoshanit Ohad, Dalit Vaizel-Ohayon, Meir Rom, Joseph Guttman, Diego Berger, Valeria Kravitz, Shlomo Pilo, Zohar Huberman, Yechezkel Kashi, Efrat Rorman, and et al. Microbial source tracking in adjacent karst springs. *Applied and Environmental Microbiology*, 81(15):5037–5047, 2015. doi: 10.1128/aem.00855-15.

[13] Dan Wang, Sarah S. Silkie, Kara L. Nelson, and Stefan Wuertz. Estimating true human and animal host source contribution in quantitative microbial source tracking using the monte carlo method. *Water Research*, 44(16):4760–4775, 2010. doi: 10.1016/j.watres.2010.07.076.

[14] Alexandria Graves, Charles Hagedorn, A Brooks, R.L. Hagedorn, and Emily Martin. Microbial source tracking in a rural watershed dominated by cattle. 41:3729–3739, 09 2007.

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning: with applications in R*. Springer, 2017.

[16] Hadley Wickham and Garrett Grolemund. *R for data science: import, tidy, transform, visualize and model data*. OReilly, 2017.

[17] The comprehensive r archive network. URL `https://cran.r-project.org/`. Accessed: 15 February 2018.

[18] Electricity usage of a laptop, notebook or netbook. URL `http://energyusecalculator.com/electricity_laptop.htm`. Accessed: 20 March 2018.

[19] Barcelona power station. URL `https://en.wikipedia.org/wiki/Barcelona_power_station`. Accessed: 20 March 2018.

[20] How much co2 emissions per kwh of electricity? URL `https://carbonpositivelife.com/co2-per-kwh-of-electricity/`. Accessed: 20 March 2018.

[21] Curse of dimensionality. URL `https://en.wikipedia.org/wiki/Curse_of_dimensionality`. Accessed: 10 April 2018.

[22] Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers, 2001.

[23] Robert Bain, Jamie Bartram, Mark Elliott, Robert Matthews, Lanakila Mcmahan, Rosalind Tung, Patty Chuang, and Stephen Gundry. A summary catalogue of microbial drinking water tests for low and medium resource settings. *International Journal of Environmental Research and Public Health*, 9(5):1609–1625, Apr 2012. doi: 10.3390/ijerph9051609.

[24] Hyperplane. URL https://en.wikipedia.org/wiki/Hyperplane. Accessed: 23 February 2018.

[25] Support vector machines. URL https://www.stat.berkeley.edu/~arturof/Teaching/EE127/Notes/support_vector_machines.pdf. Accessed: 21 April 2018.

[26] Decision tree learning. URL https://en.wikipedia.org/wiki/Decision_tree_learning. Accessed: 28 March 2018.

[27] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[28] Joyce Chepkemoi. The world's most saline bodies of water, Jan 2017. URL https://www.worldatlas.com/articles/the-world-s-most-saline-bodies-of-water.html. Accessed: 21 March 2018.

[29] What is a box plot and when to use it. URL https://chartio.com/resources/tutorials/what-is-a-box-plot/. Accessed: 20 April 2018.

[30] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, Jan 2002. ISSN 1573-0565. doi: 10.1023/A:1012487302797. URL https://doi.org/10.1023/A:1012487302797.

[31] Alexandros Karatzoglou, Alexandros Smola, Kurt Hornik, and Achim Zeileis. kernlab - an s4 package for kernel methods in r. *Journal of Statistical Software, Articles*, 11(9):1–20, 2004. ISSN 1548-7660. doi: 10.18637/jss.v011.i09. URL https://www.jstatsoft.org/v011/i09.

[32] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[33] F1 score. URL https://en.wikipedia.org/wiki/F1_score. Accessed: 15 June 2018.